1991

Solution Adaptive Grid Procedures for High-Speed Parabolic Flows.

Albert Destrehan Harvey III
Louisiana State University and Agricultural & Mechanical College

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9219540

Solution-adaptive grid procedures for high-speed parabolic flows

Harvey, Albert Destrehan, III, Ph.D.

The Louisiana State University and Agricultural and Mechanical Col., 1991

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

SOLUTION-ADAPTIVE GRID PROCEDURES FOR HIGH-SPEED PARABOLIC FLOWS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Mechanical Engineering

by
Albert Destrehan Harvey III
B.S.M.E., Louisiana State University, 1987
December 1991

Acknowledgements

I would like to express my sincere appreciation to Professor Sumanta Acharya for his continued assistance and motivation throughout the completion of this work. His constructive review of this dissertation was instrumental in its completion.

I also wish to acknowledge the financial support provided by the Applied Computational Fluids Branch at NASA Ames Research Center under Joint Interchanges NCA2-326 and NCA2-552. The technical expertise of Dr. Scott Lawrence, who served as technical monitor, is most gratefully appreciated. I would also like to thank Dr. Terry Holst, Branch Chief of the Applied Computational Fluids Branch, for allowing me this special research opportunity and for the use of the computational facilities at Ames Research Center.

Lastly, but also most importantly, I would like to thank my wife, Janet, whose everlasting love, patience and understanding during the last four years has been the key to my motivation and success.

Table of Contents

Page

Acknowledgements ii

List of Figures vi

Abstract xii

Chapter

1. Introduction 1

1.1 Statement of Research Objectives 2

1.2 Previous Work on Adaptive Grids 3

1.3 Outline of the Dissertation 5

2. Flow Algorithm 8

2.1 Governing Equations 9

2.2 Discretization Method 14

2.3 Streamwise Numerical Flux 15

2.4 First-Order Upwind Scheme 16

2.4.1 Approximate Riemann Solvers 17

2.4.2 Steady Riemann Solvers 19

2.5 Second-Order Upwind Scheme 22

2.6 Implicit Solution Procedure 25

2.7 Grid Generation 26

3. Two-Dimensional Adaptive Grid Procedure 29

3.1 Adaptation Strategy 30

3.2 Selection of Weighting Functions 36

3.3 Grid-Fitting Algorithm 38

Chapter *Page*

4. Two-Dimensional Applications 41

4.1 Hypersonic Compression Corner 42

4.2 Hypersonic Inlet 52

4.3 Cone-Cylinder 58

4.4 Comparison of Computational Effort 62

4.5 Weighting Function Comparisons 62

5. Three-Dimensional Adaptive Grid Procedure 68

5.1 Three-Dimensional Methodology 69

5.2 Calculation of Torsional Parameters 71

5.3 Determination of Reference Locations 75

5.4 Adaptation Boundaries 80

5.5 Algorithmic Procedure 81

6. Three-Dimensional Applications: Flow over Cones 83

6.1 Right-Circular Cone at Incidence 84

6.2 Grids and Contour Plots 87

6.2.1 Yaw Angle of 8° 87

6.2.2 Yaw Angle of 16° 91

6.2.3 Yaw Angle of 24° 96

6.3 Pressure and Heat Transfer Comparisons 102

7. Application to Complex Geometries 109

7.1 Algorithm Improvements 110

7.2 Torsional Dependence, Re-Visited 112

7.3 Torsional Force Terms 114

7.4 Three-Dimensional Grid-Fitting Procedure 117

7.5 Ames Allbody Vehicle 122

Chapter *Page*

7.6 McDonnell Douglas Generic Option 142

8. Application to Chemical Nonequilibrium Flows 150

8.1 Overview of Chemistry Model 151

8.2 Solution Procedure 154

8.3 Flow over a Blunt Cone 156

9. Concluding Remarks 168

9.1 Overview 169

9.2 Summary of Present Contributions 172

9.3 Comments on Further Research 175

Bibliography 177

Appendices

A. Eigenvalues and Eigenvectors of the D Matrix 181

B. Derivation of Inviscid Flux Jacobians 182

C. Viscous Flux Jacobians 185

D. Calculation of Weighting Function Exponent 187

E. Letter of Permission 189

Vita 190

List of Figures

Fig.	*Title*	*Page*
2.1	Illustration of coordinate transformation.	12
2.2	Finite volume cell geometry.	18
2.3	Evaluation of interface fluxes.	18
3.1	Adaptive mesh geometry.	32
3.2	Tensional and torsional spring system.	32
4.1	Compression corner geometry.	43
4.2	Geometry and computational grids for hypersonic flow over a compression corner for three test cases. (a) Without solution-adaptive gridding, (b) with solution-adaptive gridding and, (c) with adaptive gridding and grid-fitting. Every second grid line is shown in the normal direction for all three grids. All dimensions are in meters.	44
4.3	Computational grids in the vicinity of the shock interaction region. (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting. All dimensions are shown in meters.	46
4.4	Pressure contours in the shock interaction region. (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting.	47
4.5	Mach number contours in the shock interaction region.& (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting.	49
4.6	Comparison of pressure profiles ($x = 0.7m$).	50
4.7	Comparison of wall pressure coefficients.	51

Fig.	*Title*	*Page*
4.8	Comparison of heat transfer coefficients.	51
4.9	Hypersonic geometry.	53
4.10	Computational grid for planar-inlet configuration.	55
4.11	Comparison of computed pressure contours. Planar inlet (a) unadapted and, (b) adapted.	56
4.12	Pressure profiles at ($x = 0.9m$).	57
4.13	Cone-cylinder geometry.	59
4.14	Adapted grid for cone-cylinder.	60
4.15	Comparison of pressure signatures.	60
4.16	Comparison of weighting functions for compression ramp test case. ($M_\infty = 14.1$, $Re_{\infty,L} = 1.04 \times 10^5$, $x = .7m$, Δs_{MAX}=4.5, Δs_{MIN}=.15)	65
5.1	Three-dimensional adaptive mesh geometry.	70
5.2	Torsion vector geometry.	76
5.3	Determination of torsion vector locations.	79
6.1	Conical flow geometry.	86
6.2	Sectional view of solution-adapted computational grid ($\alpha = 8°$).	88
6.3	Comparison of cross-flow computational grids at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid ($\alpha = 8°$).	89
6.4	Comparison of Mach number contours at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid. ($\alpha = 8°$).	90
6.5	Sectional view of solution-adapted computational grid ($\alpha = 16°$).	92
6.6	Comparison of cross-flow computational grids at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid ($\alpha = 16°$).	94
6.7	Comparison of Mach number contours at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid. ($\alpha = 16°$).	95

Fig.	*Title*	*Page*
6.8	Comparison of cross-flow computational grids at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid ($\alpha = 24°$).	97
6.9	Comparison of Mach number contours at $x = 0.1016m$. Left-half, fixed grid; right-half, solution-adapted grid. ($\alpha = 24°$).	98
6.10	Comparison of cross-flow computational grids at $x = 0.1016m$. Left-half, adapted in radial direction only; right-half, adapted in circumferential direction only ($\alpha = 24°$).	100
6.11	Comparison of Mach number contours at $x = 0.1016m$. Left-half, adapted in radial direction only; right-half, adapted in circumferential direction only ($\alpha = 24°$).	103
6.12	Comparison of pitot-pressure profiles at leeward meridian. (x= .1016m)	104
6.13	Circumferential surface pressure distributions at $x = 0.1016m$.	106
6.14	Comparison of Mach number profiles at leeward meridian. (x= .1016m)	107
6.15	Circumferential heat transfer distributions at $x = 0.1016m$.	108
7.1	Adaptive mesh geometry.	113
7.2	Schematic of grid-fitting procedure.	119
7.3	Downstream projection of adapted grid plane.	121
7.4	Allbody hypersonic vehicle geometry.	123
7.5.	Sectional view of solution-adapted and grid-fitted computational grid. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, 151×90 cross-flow grid)	125
7.6	Comparison of cross-flow computational grid planes; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$, 151×90 grids)	127

Fig.	*Title*	*Page*
7.7	Comparison of computed Mach number contours; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)	129
7.8	Comparison of computed pressure contours; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)	130
7.9.	Sectional view of solution-adapted and grid-fitted computational grid. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, 151×90 cross-flow grid)	131
7.10	Comparison of cross-flow computational grid planes; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$, 151×90 grids)	132
7.11	Comparison of computed Mach number contours; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)	133
7.12	Comparison of computed pressure contours; left-half, fixed grid; right-half, solution-adapted and grid fitted. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)	134
7.13	Comparison of cross-flow computational grid planes; left-half, solution-adapted; right-half, solution-adapted and grid fitted. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$, 151×90 grids)	136

Fig. *Title* *Page*

7.14 Comparison of grids and Mach number contours near cross-flow stagnation point. (a) fixed grid, (b) adapted, w/o grid-fitting (c) adapted, grid fitted. ($\alpha = 5°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$, 151×90 grids) 137

7.15 Comparison of pitot pressure profiles. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$) 139

7.16 Comparison of pitot pressure profiles. ($M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .8$) 140

7.17 Comparison of symmetry plane surface pressure distributions. ($M_\infty = 7.4$,$Re_{\infty,L} = 15 \times 10^6$) 141

7.18 Surface grid for McDonnell Douglas Generic Option Vehicle. 143

7.19 Illustration of solution-adapted and grid-fitted grid and Mach contours for McDonnell Douglas generic option vehicle, (a) grid, (b) solution-adapted, grid-fitted (c) fixed grid predictions. ($\alpha = 0°$, $M_\infty = 11.35$, $Re_\infty = 9.14 \times 10^5/in$,$65 \times 65$ cross-flow grid) 144

7.20 Comparison of grids and Mach number contours near maximum spanwise location. (a) fixed grid, (b) adapted, w/o grid-fitting, (c) adapted and grid-fitted. ($\alpha = 0°$, $M_\infty = 11.35$, $Re_\infty = 9.14 \times 10^5/in$, $x = 14.68in$, 65×65 grid) 146

7.21 Comparison of computational grids at $x = 17 - 28in$ for the McDonnell Douglas generic option vehicle. (a) fixed grid, (b) solution-adapted. ($\alpha = 0°$, $M_\infty = 11.35$, $Re_\infty = 9.14 \times 10^5/in$, 65×65 crossflow grid) 147

Fig.	*Title*	*Page*
7.22	Comparison of Mach number contours at $x = 17 - 28in$ for the McDonnell Douglas generic option vehicle. (a) fixed grid, (b) solution-adapted. ($\alpha = 0°$, $M_\infty = 11.35$, $Re_\infty = 9.14 \times 10^5/in$, 65×65 crossflow grid)	148
8.1	Solution-adapted and grid-fitted grid. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 grid)	158
8.2	Solution-adapted grid and outer grid surface for blunt cone near start of marching procedure showing transition from fixed to adapted grid.	159
8.3	Mach number contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 ADAPTED grid)	160
8.4	Mach number contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 FIXED grid)	161
8.5	Atomic oxygen contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 ADAPTED grid)	162
8.6	Atomic oxygen contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 FIXED grid)	163
8.7	Pressure contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 ADAPTED grid)	164
8.8	Pressure contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×65 FIXED grid)	165
8.9	Mach number contours. ($\alpha = 5°$, $M_\infty = 15$, $Re_{\infty,L} = 1.33 \times 10^5$, $T_\infty = 227K$, 66×85 FIXED grid)	166

Abstract

A solution-adaptive grid procedure based on an error equi-distribution scheme is developed and applied to a Parabolized Navier-Stokes solver. The adaptation scheme re-distributes grid points line-by-line, with grid point motion controlled by forces analogous to tensional and torsional spring forces with the tensional spring force at a point proportional to the error estimate. Where the error estimates are large, the tensional forces become high and drive grid points to cluster in these high error estimate regions. Torsional terms are functions of the grid point positions along neighboring grid lines and provide grid smoothness and orthogonality. A method of selecting error estimates or weighting functions is introduced which involves normalizing a combination of flow-field gradients and curvature of a number of dependent variables and then selecting the largest at each point. The resulting grid is therefore adapted based on each independent variable. A grid-fitting scheme is introduced for external flows in which the farfield extent of the computational grid is adjusted during the course of the calculation procedure in order to align the grid with flow-field structure and to minimize the number of grid points in the freestream.

The adaptive grid method is tested on a variety of two-dimensional problems which include the hypersonic flow over a leading-edge compression ramp, the hypersonic flow through a cowl-type inlet configuration and the axi-symmetric flow over a cone-cylinder geometry. For each case, flow-field results computed using the solution-adaptive algorithm show significant improvements over results obtained using a fixed grid.

Three-dimensional applications solved numerically include, (a) hypersonic flow over a right circular cone at various angles of freestream incidence, (b) flow over the NASA Ames allbody vehicle, (c) flow over the McDonnell Douglas generic option vehicle and (d) the chemically reacting flow over a blunt cone at incidence.

Again, numerical results using the solution-adaptive grid scheme show significant improvements in flow-field resolution. Pitot-pressure predictions obtained using the solution-adaptive technique show improved agreement with experiment over those obtained using a conventional fixed grid.

CHAPTER 1

Introduction

The advent of the high-speed digital computer has given rise to a new method for investigating aerothermodynamic phenomena which is commonly referred to as *Computational Fluid Dynamics* (CFD). This method involves replacing the continuous equations of fluid motion with a set of algebraic equations for flow properties at a finite number of discrete grid points. In recent years, this method of fluid flow investigation has become an important tool, especially in the realm of hypersonic flow where experimental procedures are often difficult due to extreme conditions. The use of CFD is expected to be play a crucial role in the design of high-speed aircraft such as the National Aerospace Plane (NASP) where the accurate prediction of aerothermal loads is extremely important.

1.1 Statement of Research Objectives

The main objective of this research project is to develop an appropriate solution-adaptive grid procedure for use with a Parabolized Navier-Stokes (PNS) solver, specifically the NASA *UPS* flow solver. The *UPS* flow solver employs a spatial marching technique to yield solutions to flow-field variables at high speeds (supersonic–hypersonic). The use of solution-adaptive grid techniques allows for improved flow-field solutions by re-clustering grid points into regions of high solution error.

Any numerical differencing scheme is, to some extent, dependent on the geometrical representation of the region of interest. The region is usually represented by a finite collection of grid points at which physical variables are to be evaluated. The governing partial differential equations for the physical variables at each grid point are approximated by an algebraic equation and the difference between the exact solution and the numerical solution, called the truncation error, is proportional to $(\Delta s)^n f(\phi, \phi', \cdots, \phi^n)$, where n is the order of the numerical approximation. It therefore follows that as the number of available grid points increases, Δs decreases and therefore the accuracy of the algebraic approximation of the differential equations is improved. To equally distribute an overall measure of the truncation error in the domain, the grid spacing Δs, should be reduced where the measure $f(\phi, \phi', \cdots, \phi^n)$ is large. However, this error measure, hereafter called the error measure or weighting function, can only be determined in an *a posteri* manner.

The objective of this research is to devise a mechanism whereby, as the numerical solution evolves, the grid spacing in regions of high truncation error is reduced, so that the corresponding truncation error is also reduced. In this manner the resulting numerical solution is improved by a refinement in the representation of the geometry based on the physical structure of the evolving flow-field. This

concept is called solution-adaptive gridding, and in this dissertation, solution-adaptive gridding is developed and applied to a PNS solver.

1.2 Previous Work on Adaptive Grids

The search for more accurate and computer-efficient solutions to complex problems in fluid flow and heat transfer has led to the utilization of improved discretization methods. In many such problems, there occur regions in the physical domain where the dependent variables exhibit large changes in gradient and/or curvature. For supersonic flow-fields, these regions could include shock waves, expansion fans and contact surfaces, as well as boundary-layers which, when present simultaneously, can produce physically complex interactions that are difficult to simulate numerically without strategic grid point placement. In most cases, the locations of these regions are not known *a priori*, and hence, the initial distribution of grid points is unsatisfactory. What is needed is an adaptive grid method for re-distributing the grid points as the solution evolves. This re-distribution should sufficiently cluster points in high gradient regions while not completely devoiding other regions of grid points.

Numerous studies on adaptive grid techniques are presently available. Thompson (1985a) and Hawken (1985) independently provide complete surveys of the most widely accepted methods. As noted by Thompson, most adaptive grid procedures attempt to equally distribute some measure of the solution error; however, each differs in its individual approach. Moukalled (1987) also provides a rigorous review of presently available grid adaptation techniques.

Adaptive grid methods can be categorized into two basic techniques; those that incorporate a global grid point re-distribution and those that move points only in certain pre-determined regions. In the global re-distribution process, all available grid points participate in the adaptation process. A measure of the solution error is equally distributed throughout the entire region of interest.

Present techniques employing this approach are numerous. Dwyer (1984), Dwyer et al. (1980) and Nakahashi and Deiwert (1986, 1988) are a few representative examples.

In the local refinement technique, the most popular approach has been to uniformly sub-divide important regions. In the localized adaptation approach such as those of Moukalled (1987) and Berger and Jameson (1985), additional grid points are added to 'important' regions. A re-distribution of grid points then takes place in these refined regions while the regions of less importance remain unchanged. In the method of Kallinderis (1989), refinement of the grid takes place in these pre-determined important regions but no re-distribution takes place.

The most popular global re-distribution approach has been to maintain a constant product of a weighting function, which is proportional to an error measure, and the grid interval throughout the solution domain. Dwyer et al. (1980, 1984) used a linear combination of the gradients and curvature of a dependent flow variable as the error measure. Rai and Anderson (1982) and Greenberg (1983) have used an attraction/repulsion method where grid points possessing weighting functions larger than some average value attract each other while those points with values less than the average repel each other. Gnoffo (1982) introduced a method analogous to a system of tension springs set between grid points whose spring constants are functions of an error measure. Nakahashi and Deiwert (1984, 1985) extended the method to include torsion spring forces that relate grid point positions along neighboring lines and thus provide for grid smoothness. This adaptive grid method employing the tension and torsion spring concept was used to significantly enhance the solution to two-dimensional airfoil flows [Nakahashi and Deiwert (1984, 1985)] This method has also been used by Davies and Venkatapathy (1989) in the development of the two-dimensional *SAGE* algorithm. Nakahashi and Deiwert (1986) extended the concept to include multiple torsion

springs for application to three-dimensions. Djomehri and Deiwert (1988) have also successfully applied the method to three-dimensional geometries.

Variational methods have also been used as a method of grid adaptation. This approach involves the minimization of an integral whose integrand is a function of some error estimate of the evolving solution. Thompson (1985a,b) has shown that this minimization process is equivalent to the solution of a set of elliptic partial differential equations (Poisson's equations) with the non-homogeneous terms proportional to the error measure. The method of Gnoffo (1982) can be described in terms of variational methods as the minimization of the energy in a system of springs set between grid points. In the variational approach of Brackbill (1982), measures of grid orthogonality, straightness and grid spacing size are optimized.

The objective of this study, as noted earlier, is to develop and couple an efficient solution-adaptive grid scheme to a parabolic space-marching flow solver. Solutions to spatially parabolic equations are obtained by marching in space rather than time and therefore are obtained much more efficiently than solutions to the time-dependent Navier-Stokes equations.

1.3 Outline of the Dissertation

This dissertation describes the development of a three-dimensional solution-adaptive grid procedure for use in a parabolized Navier-Stokes (PNS) solver. This development evolves throughout the following chapters, starting out as a two-dimensional adaptive grid method which is applied to various two-dimensional hypersonic flow-fields. The algorithm is then extended to three-dimensions for application to relatively simple geometries such as a right-circular cone. In the final chapters, the three-dimensional algorithm is extended for application to more complex (and general) three-dimensional configurations. The resulting three-dimensional solution-adaptive PNS solver is then used to obtain numerical predictions for the hypersonic flow over two generic aircraft configurations.

The present flow solver employs a spatial marching procedure for the numerical integration of the steady parabolized Navier-Stokes equations. Chapter 2 outlines this procedure as well as the assumptions and limitations used in the derivation of the parabolized Navier-Stokes equations.

Chapter 3 describes the two-dimensional development of the solution-adaptive grid algorithm. This chapter contains the procedure whereby the truncation error is estimated. This error estimate or weighting function is used to drive the movement of grid points. Also included in Chapter 3 is a two-dimensional grid-fitting scheme which monitors the location of the outer shock during the course of the solution process and adjusts the outer boundary to coincide with this location. This approach maintains a maximum number of grid points inside the region of interest.

A variety of applications of the two-dimensional solution-adaptive grid scheme are presented in Chapter 4. Numerical predictions obtained using the solution-adaptive grid method are shown to be superior to those obtained using conventional fixed grids. Where possible, numerical predictions are compared with experimental results. Also included in Chapter 4 is a study illustrating a few weighting function selection techniques.

In Chapter 5, the two-dimensional solution-adaptive grid method developed in Chapter 3 is extended to three-dimensions. This includes procedures for adapting the grid in both cross-flow directions and for projecting the adapted grid downstream to generate a preliminary cross-flow grid plane at the next marching station.

In Chapter 6, the hypersonic flow over a right-circular cone at three different yaw angles is studied to demonstrate the improvements that can be obtained with the three-dimensional solution-adaptive grid procedure. For each angle of incidence investigated, pitot pressure profiles computed using the solution-adaptive

algorithm show improved agreement with experiments over results obtained using a conventional fixed grid.

In Chapter 7, the solution-adaptive marching procedure is further extended for application to more general three-dimensional geometries. These applications require extensive modifications to the grid adaptation procedures of the preceding chapters. A three-dimensional grid-fitting scheme is also developed which monitors the position of freestream conditions and adjusts the grid to maintain a maximum grid point density inside the region of interest. The resulting algorithm is applied to the hypersonic flow about two generic aircraft configurations. The first of these is an all-body type geometry with elliptical cross sections and is studied at various angles of attack. The second geometry is a generic blended-wing-body design. The results demonstrate the improvements in flow-field resolution obtainable with the solution-adaptive marching procedure over the conventional fixed grid technique. Computed pitot pressure profiles and surface pressure distributions for the solution-adaptive predictions are compared with results obtained using a fixed grid and with experimental results.

In Chapter 8, an application of the solution-adaptive algorithm to the nonequilibrium flow of air over a blunt cone at angle of incidence is investigated. Mach number and pressure contours resulting from the use of the adaptive grid is compared to those obtained using a fixed grid of equal size as well as one of higher grid point density.

Chapter 9 provides a summary of the dissertation and comments on some proposed future work.

CHAPTER 2

Flow Algorithm

In this section the Upwind Parabolized Navier-Stokes Solver (UPS) used in the present work is presented. This algorithm, developed by Lawrence (1987), involves the numerical integration of the parabolized Navier-Stokes equations using an upwind differencing technique to determine the inviscid fluxes. The references to be cited in this section more than adequately describe the methods and procedures employed; however, to provide a sufficient level of completeness in the present work, this chapter is devoted to describing these techniques. In some sections of this chapter, the notation of Lawrence (1984, 1987) is adopted in order to preserve the originality of the methods.

2.1 Governing Equations

The present algorithm involves the integration of the parabolized Navier-Stokes (PNS) equations, in which the spatial propagation of flow-field information is locally modeled using a steady version of Roe's scheme (Roe, 1981). The PNS equations are obtained from the steady-state Navier-Stokes equations by neglecting streamwise viscous derivatives and by extracting the portion of the streamwise pressure gradient term that is responsible for introducing ellipticity into the equations.

The Navier-Stokes equations are parabolized with respect to the streamwise coordinate direction by first making the following assumptions: 1) the flow is steady, and 2) the viscous derivatives in the streamwise direction are negligible in comparison with those in the cross-flow directions. The latter assumption is generally considered valid for high Reynolds number flows. The steady Navier-Stokes equations can be written in a strong conservative form as

$$\frac{\partial(\mathbf{E}_i - \mathbf{E}_v)}{\partial x} + \frac{\partial(\mathbf{F}_i - \mathbf{F}_v)}{\partial y} + \frac{\partial(\mathbf{G}_i - \mathbf{G}_v)}{\partial z} = 0 \tag{2.1}$$

The inviscid and viscous flux vectors, **E**, **F**, and **G** are defined below, the subscripts i and v indicating inviscid and viscous components, respectively.

$$\mathbf{E}_i = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E_t + p)u \end{pmatrix} \qquad \mathbf{E}_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{pmatrix}$$

$$\mathbf{F}_i = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E_t + p)v \end{pmatrix} \qquad \mathbf{F}_v = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y \end{pmatrix} \tag{2.2}$$

$$\mathbf{G}_i = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E_t + p)w \end{pmatrix} \qquad \mathbf{G}_v = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{pmatrix}$$

with

$$E_t = \rho[e + \frac{1}{2}(u^2 + v^2 + w^2)]$$

The viscous stress and heat transfer terms are given by

$$\tau_{xx} = \frac{2}{3}\frac{\mu}{Re_L}\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right) \qquad \tau_{xy} = \frac{\mu}{Re_L}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) \qquad q_x = \psi_t \frac{\partial T}{\partial x}$$

$$\tau_{yy} = \frac{2}{3}\frac{\mu}{Re_L}\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right) \qquad \tau_{xz} = \frac{\mu}{Re_L}\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) \qquad q_y = \psi_t \frac{\partial T}{\partial y}$$

$$\tau_{zz} = \frac{2}{3}\frac{\mu}{Re_L}\left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \qquad \tau_{yz} = \frac{\mu}{Re_L}\left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right) \qquad q_z = \psi_t \frac{\partial T}{\partial z}$$

where

$$\psi_t = \frac{\mu}{(\gamma - 1)M_\infty^2 Re_L Pr}$$

In these equations, p is the non-dimensional pressure, ρ is the density, u, v and w are the velocity components in the x, y and z directions, e is the internal energy, τ is the viscous stress, and q is the heat conduction rate. The molecular viscosity, μ, is computed using a non-dimensional form of the Sutherland law (Lawrence, 1987). The thermal conductivity is expressed in terms of the Prandtl number and the molecular viscosity and, as a result, thermal conductivity does not explicitly appear in the heat conduction terms above. The following non-dimensional quantities are used, with a tilde representing dimensional quantities.

$$x = \frac{\tilde{x}}{\tilde{L}} \qquad u = \frac{\tilde{u}}{\tilde{V}^*} \qquad \rho = \frac{\tilde{\rho}}{\tilde{\rho}_\infty} \qquad e = \frac{\tilde{e}}{\tilde{p}_\infty/\tilde{\rho}_\infty}$$

$$y = \frac{\tilde{y}}{\tilde{L}} \qquad v = \frac{\tilde{v}}{\tilde{V}^*} \qquad p = \frac{\tilde{p}}{\tilde{p}_\infty} \qquad \mu = \frac{\tilde{\mu}}{\tilde{\mu}_\infty}$$

$$z = \frac{\tilde{z}}{\tilde{L}} \qquad w = \frac{\tilde{w}}{\tilde{V}^*} \qquad T = \frac{\tilde{T}}{\tilde{T}_\infty} \qquad \tilde{V}^* = \sqrt{\tilde{p}_\infty/\tilde{\rho}_\infty}$$

where $\tilde{L}$ is a suitable reference length. The non-dimensional system of equations is closed with non-dimensional forms for the perfect gas equations of state, expressed as

$$p = (\gamma - 1)\rho e \qquad \text{and} \qquad T = \gamma M_\infty^2 p/\rho$$

The discretization of these equations over an arbitrary body is best achieved by first expressing the equations in terms of a generalized geometry-oriented system of coordinates. In this manner, integration takes place over regularly shaped unit cells (computational space), and the coordinate transformation describes a mapping from the computational space to physical space (See Fig. 2.1). In later sections we will see that grid adaptation takes place in the physical domain. This mapping is then adjusted, afterwards, to maintain uniform unit cells in the computational domain.

For the present study, the coordinate transformation of the general form

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{2.3}$$

is applied. The Jacobian, J, of the transformation is given by

$$J = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)}$$

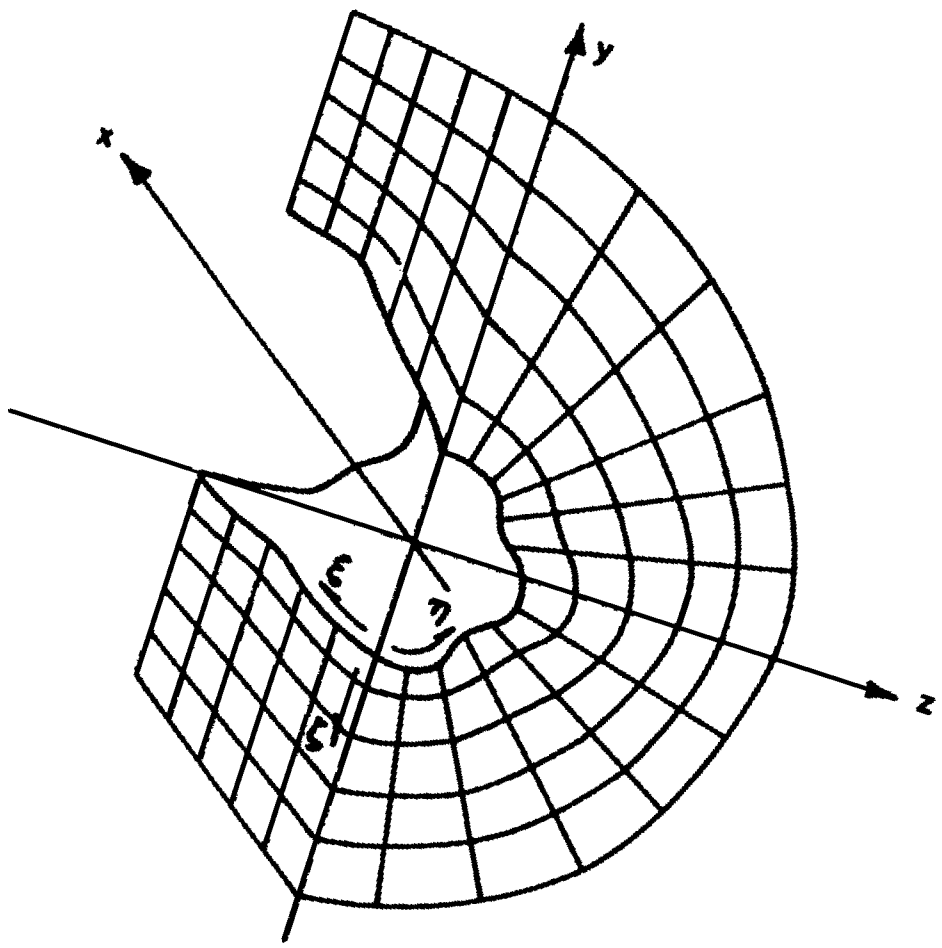
and can be calculated using

$$J^{-1} = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) - x_\eta(y_\xi z_\zeta - y_\zeta z_\xi) + x_\zeta(y_\xi z_\eta - y_\eta x_\xi)$$

The metrics are given by the following expressions

$$\begin{aligned} \left(\frac{\xi_x}{J}\right) &= y_\eta z_\zeta - y_\zeta z_\eta & \left(\frac{\eta_x}{J}\right) &= y_\zeta z_\xi - y_\xi z_\zeta & \left(\frac{\zeta_x}{J}\right) &= y_\xi z_\eta - y_\eta z_\xi \\ \left(\frac{\xi_y}{J}\right) &= x_\zeta z_\eta - x_\eta z_\zeta & \left(\frac{\eta_y}{J}\right) &= x_\xi z_\zeta - x_\zeta z_\xi & \left(\frac{\zeta_y}{J}\right) &= x_\eta z_\xi - x_\xi z_\eta \\ \left(\frac{\xi_z}{J}\right) &= x_\eta y_\zeta - x_\zeta y_\eta & \left(\frac{\eta_z}{J}\right) &= x_\zeta y_\xi - x_\xi y_\zeta & \left(\frac{\zeta_z}{J}\right) &= x_\xi y_\eta - x_\eta y_\xi \end{aligned} \tag{2.4}$$

All derivatives with respect to the Cartesian coordinate system can be evaluated with respect to the new coordinate system using the chain rule of differentiation,



COMPUTATIONAL SPACE

PHYSICAL SPACE

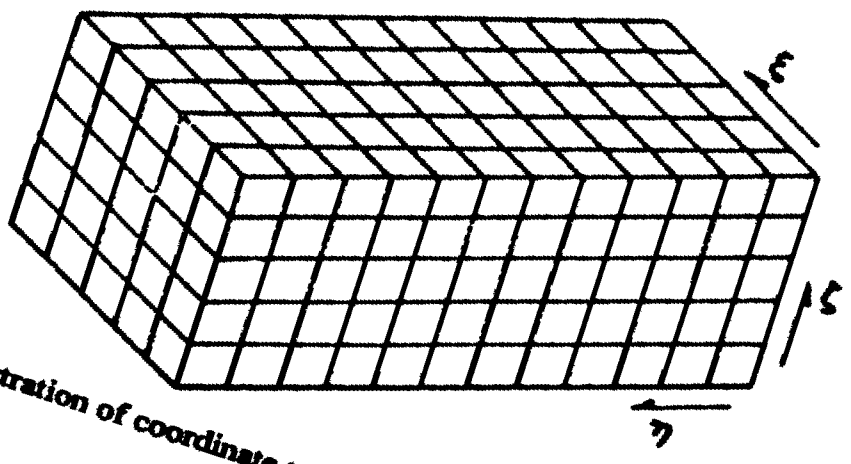


Figure 2.1: Illustration of coordinate transformation.

written here as

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta}$$
$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta}$$
$$\frac{\partial}{\partial z} = \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta}$$

After expressing the terms of Eq. (2.1) using the chain rule (See, for example Hoffmann, 1989), this equation can be expressed in terms of the generalized coordinate system as

$$\frac{\partial \bar{\mathbf{E}}}{\partial \xi} + \frac{\partial \bar{\mathbf{F}}}{\partial \eta} + \frac{\partial \bar{\mathbf{G}}}{\partial \zeta} = 0 \tag{2.5}$$

where

$$\begin{aligned}
\bar{\mathbf{E}} &= \left(\frac{\xi_x}{J}\right)\mathbf{E}_i + \left(\frac{\xi_y}{J}\right)\mathbf{F}_i + \left(\frac{\xi_z}{J}\right)\mathbf{G}_i \\
\bar{\mathbf{F}} &= \left(\frac{\eta_x}{J}\right)(\mathbf{E}_i - \mathbf{E}_v^*) + \left(\frac{\eta_y}{J}\right)(\mathbf{F}_i - \mathbf{F}_v^*) + \left(\frac{\eta_z}{J}\right)(\mathbf{G}_i - \mathbf{G}_v^*) \\
\bar{\mathbf{G}} &= \left(\frac{\zeta_x}{J}\right)(\mathbf{E}_i - \mathbf{E}_v^*) + \left(\frac{\zeta_y}{J}\right)(\mathbf{F}_i - \mathbf{F}_v^*) + \left(\frac{\zeta_z}{J}\right)(\mathbf{G}_i - \mathbf{G}_v^*)
\end{aligned} \tag{2.5a}$$

Here, ξ is the streamwise coordinate direction and η and ζ are the two cross-flow directions. The superscript asterisk on the viscous flux vectors above indicates that derivatives with respect to ξ have been omitted. Equation (2.5) is a set of mixed hyperbolic-parabolic equations with respect to the streamwise coordinate direction provided that 1) the Mach number outside the boundary layer is greater than unity, 2) the streamwise ξ-component of the velocity is greater than zero (no streamwise separation), and 3) the pressure gradient term is treated in a manner in which the ellipticity inherent in the equations is eliminated. The method of Vigneron (1987) was chosen to accomplish this task. Using this technique, the streamwise flux vector is split into two parts as follows

$$\bar{\mathbf{E}}_i = \hat{\mathbf{E}}^* + \bar{\mathbf{E}}^P \tag{2.6}$$

where

$$\hat{\mathbf{E}}^* = \begin{pmatrix} \rho\hat{U} \\ \rho u\hat{U} + (\frac{\xi_x}{J})\omega p \\ \rho v\hat{U} + (\frac{\xi_y}{J})\omega p \\ \rho w\hat{U} + (\frac{\xi_z}{J})\omega p \\ (E_t + p)\hat{U} \end{pmatrix} \qquad \hat{\mathbf{E}}^p = \begin{pmatrix} 0 \\ (\frac{\xi_x}{J})(1-\omega)p \\ (\frac{\xi_y}{J})(1-\omega)p \\ (\frac{\xi_z}{J})(1-\omega)p \\ 0 \end{pmatrix}$$

and

$$\hat{U} = \left(\frac{\xi_x}{J}\right)u + \left(\frac{\xi_y}{J}\right)v + \left(\frac{\xi_z}{J}\right)w$$

After substitution of Eq. (2.6) into Eq. (2.5), the resulting equations are referred to as the Parabolized Navier-Stokes (PNS) equations and can be written as

$$\frac{\partial\hat{\mathbf{E}}^*}{\partial\xi} + \frac{\partial\hat{\mathbf{E}}^p}{\partial\xi} + \frac{\partial\bar{\mathbf{F}}}{\partial\eta} + \frac{\partial\bar{\mathbf{G}}}{\partial\zeta} = 0 \tag{2.7}$$

If Eq. (2.7) is subjected to an eigenvalue analysis, it can be shown (Vigneron, 1987) that this system is hyperbolic-parabolic in the ξ-direction provided that ω is specified according to the relation

$$\omega = \min\left[1, \frac{\sigma\gamma M_\xi^2}{1 + (\gamma-1)M_\xi^2}\right] \tag{2.8}$$

where M_ξ is the Mach number in the ξ-direction and σ is a safety factor.

2.2 Discretization Method

The first step in the discretization of Eq. (2.7) is to divide the region of interest into small but finite hexahedrons such as the one of Fig. 2.2. Since the present algorithm employs a space-marching scheme, the complete region is discretized by successively adding cross-flow grid surfaces a distance $\Delta\xi$ downstream of the current marching station. The nth and $(n+1)$th (η,ζ) grid surfaces (n is the index of the ξ-coordinate) make up a computational slab of thickness $\Delta\xi$. Integration of flow-field properties across this slab results in new flow properties at the $(n+1)$th surface. Using the terminology of Lawrence (1987), the vertices of each cell are referred to as primary grid points, each of which is connected

by straight line segments. For accounting purposes, integer values k and l are assigned to the η and ζ positions, respectively, of secondary grid points (locations of physical properties), and half-integer values of k and l are used to denote cross-flow cell faces (See Fig. 2.2).

If it is assumed that the fluxes at the interfaces of the grid cell of Fig. 2.2 remain constant and that cell-face area vectors $d\mathbf{S}$ are oriented in the positive coordinate directions, Eq. (2.5) may be expressed for a single grid cell as

$$\begin{aligned} &\bar{\mathbf{E}}^{n+1}_{k,l} \cdot d\mathbf{S}^{n+1}_{k,l} + \bar{\mathbf{F}}^{n+\frac{1}{2}}_{k+\frac{1}{2},l} \cdot d\mathbf{S}^{n+\frac{1}{2}}_{k+\frac{1}{2},l} + \bar{\mathbf{G}}^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} \cdot d\mathbf{S}^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} \\ &- \bar{\mathbf{E}}^{n}_{k,l} \cdot d\mathbf{S}^{n}_{k,l} - \bar{\mathbf{F}}^{n+\frac{1}{2}}_{k-\frac{1}{2},l} \cdot d\mathbf{S}^{n+\frac{1}{2}}_{k-\frac{1}{2},l} - \bar{\mathbf{G}}^{n+\frac{1}{2}}_{k,l-\frac{1}{2}} \cdot d\mathbf{S}^{n+\frac{1}{2}}_{k,l-\frac{1}{2}} = 0 \end{aligned} \tag{2.9}$$

where the indexing of the cell faces of Fig. (2.2) is as follows:

$$\begin{aligned} d\mathbf{S}_{PQRS} &\equiv d\mathbf{S}^{n}_{k,l} & d\mathbf{S}_{TUVW} &\equiv d\mathbf{S}^{n+1}_{k,l} \\ d\mathbf{S}_{RSWV} &\equiv d\mathbf{S}^{n+\frac{1}{2}}_{k-\frac{1}{2},l} & d\mathbf{S}_{PQUT} &\equiv d\mathbf{S}^{n+\frac{1}{2}}_{k+\frac{1}{2},l} \\ d\mathbf{S}_{PTWS} &\equiv d\mathbf{S}^{n+\frac{1}{2}}_{k,l-\frac{1}{2}} & d\mathbf{S}_{QUVR} &\equiv d\mathbf{S}^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} \end{aligned}$$

Neglecting streamwise viscous derivatives, Eq. (2.9) can be re-written as

$$\begin{aligned} &(\hat{\mathbf{E}}_i)^{n+1}_{k,l} + (\hat{\mathbf{F}}_i - \hat{\mathbf{F}}^*_v)^{n+\frac{1}{2}}_{k+\frac{1}{2},l} + (\hat{\mathbf{G}}_i - \hat{\mathbf{G}}^*_v)^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} \\ &-(\hat{\mathbf{E}}_i)^{n}_{k,l} - (\hat{\mathbf{F}}_i - \hat{\mathbf{F}}^*_v)^{n+\frac{1}{2}}_{k-\frac{1}{2},l} - (\hat{\mathbf{G}}_i - \hat{\mathbf{G}}^*_v)^{n+\frac{1}{2}}_{k,l-\frac{1}{2}} = 0 \end{aligned} \tag{2.10}$$

where the hatted vectors represent numerical approximations to cell face fluxes and the superscript asterisks indicate that the ξ-derivatives have been eliminated. The forms of the numerical fluxes are similar to those of Eq. (2.5) with the metric quantities replaced by components of cell-face area vectors. For more on cell face definition see Lawrence (1987).

2.3 Streamwise Numerical Flux

As stated earlier, the method of Vigneron (1987) is used to suppress the ellipticity that is inherent in the boundary layer of the flow. This is done with the substitution

$$(\hat{\mathbf{E}}_i)^{n}_{k,l} = \hat{\mathbf{E}}^{*}(d\mathbf{S}^{n}_{k,l}, \mathbf{U}^{n}_{k,l}) + \hat{\mathbf{E}}^{p}(d\mathbf{S}^{n}_{k,l}, \mathbf{U}^{n-1}_{k,l})$$

where the forms of $\hat{\mathbf{E}}^*$ and $\hat{\mathbf{E}}^p$ are given by Eq. (2.6), and $d\mathbf{S}^n_{k,l}$ and $d\mathbf{U}^n_{k,l}$ indicate the location where the geometry and physical variables, respectively, are evaluated.

In order to avoid the difficulty of extracting the required flow properties from the flux vector $\hat{\mathbf{E}}^*$, a change is made in dependent variable from $\hat{\mathbf{E}}^*$ to the vector of conserved variables $\mathbf{U}$, using the following linearization

$$\hat{\mathbf{E}}^*(d\mathbf{S}^n, \mathbf{U}^n) = \hat{A}^{*^{n-1}}\mathbf{U}^n$$

where

$$\mathbf{U} = [\rho, \rho u, \rho v, \rho w, E_t]^T \qquad \text{and} \qquad \hat{A}^{*^{n-1}} = \frac{\partial \hat{\mathbf{E}}^*(d\mathbf{S}^n, \mathbf{U}^{n-1})}{\partial \mathbf{U}^{n-1}}$$

After substitution, the first and fourth terms of Eq.(2.10) become

$$(\hat{\mathbf{E}}_i)^{n+1}_{k,l} = \hat{A}^{*^n}_{k,l}\mathbf{U}^{n+1}_{k,l} + \hat{\mathbf{E}}^p(d\mathbf{S}^{n+1}_{k,l}, \mathbf{U}^n_{k,l})$$

$$(\hat{\mathbf{E}}_i)^n_{k,l} = \hat{A}^{*^{n-1}}_{k,l}\mathbf{U}^n_{k,l} + \hat{\mathbf{E}}^p(d\mathbf{S}^n_{k,l}, \mathbf{U}^{n-1}_{k,l})$$

After inserting these expressions into Eq. (2.10), rearranging and subtracting $\hat{A}^{*^n}_{k,l}\mathbf{U}^n_{k,l}$ from both sides, the discretized conservation law takes the form

$$\hat{A}^{*^n}_{k,l}\delta\mathbf{U}_{k,l} = -(\hat{A}^{*^n}_{k,l} - \hat{A}^{*^{n-1}}_{k,l})\mathbf{U}^n_{k,l} - [(\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)^{n+\frac{1}{2}}_{k+\frac{1}{2},l} - (\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)^{n+\frac{1}{2}}_{k-\frac{1}{2},l}]$$
$$-[(\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} - (\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)^{n+\frac{1}{2}}_{k,l-\frac{1}{2}}] - [\hat{\mathbf{E}}^p(d\mathbf{S}^{n+1}_{k,l}, \mathbf{U}^n_{k,l}) - \hat{\mathbf{E}}^p(d\mathbf{S}^n_{k,l}, \mathbf{U}^{n-1}_{k,l})] \tag{2.11}$$

where

$$\delta\mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^n$$

2.4 First-Order Upwind Scheme

Hyperbolic partial differential equations, such as Eqs. (2.7), are characterized by the existence of a limited domain of dependence. The solution at a point in the flow-field does not depend on every other point in the domain. Flow-field

information only travels in certain characteristic directions. Numerical techniques used for solving hyperbolic equations are usually enhanced by insuring that information is propagated only in the direction specified by the governing equation. This can be accomplished using an upwind scheme, the use of which can improve the stability of the resulting algorithm to the point where added numerical smoothing is not necessary.

The present flow solver, developed by Lawrence (1984), is based on a steady version of Roe's flux-difference split scheme (Roe, 1983) for application to the present space-marching technique. As demonstrated by van Leer et al. (1987), fewer points are required to resolve shock waves using a first-order Roe scheme than are required by a second order flux-vector split scheme (See, for example, Steger and Warming, 1981).

2.4.1 Approximate Riemann Solvers

The Riemann problem of Roe can most easily be described by considering the one-dimensional inviscid flow of a compressible gas. The conservation laws of mass, momentum and energy for such a flow can be expressed in a form analogous to Eq. (2.1) as

$$\mathbf{w}_t + \mathbf{F}_x = 0 \tag{2.12}$$

where the vectors $\mathbf{w}$ and $\mathbf{F}$ are of the form

$$\mathbf{w} = [\rho, \rho u, \rho e]^T \quad \text{and} \quad \mathbf{F} = [\rho u, \rho u^2 + p, (\rho e + p)u]^T$$

An approximate form of Eq. (2.12) with $\mathbf{F}$ expressed as some linear function of $\mathbf{w}$ can also be written as

$$\mathbf{w}_t + D\mathbf{w}_x = 0$$

where D is a constant Jacobian matrix $\partial\mathbf{F}/\partial\mathbf{w}$. Given any two states $\mathbf{w}_L$ and $\mathbf{w}_R$ to the left and right of the cell interface, the flux difference across the interface

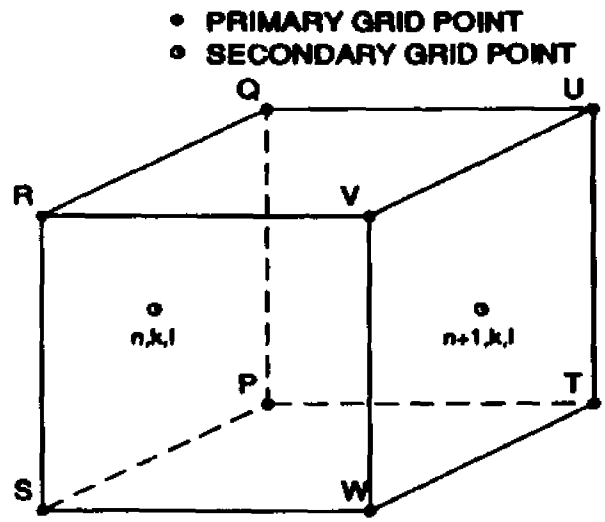


Figure 2.2: Finite volume cell geometry.

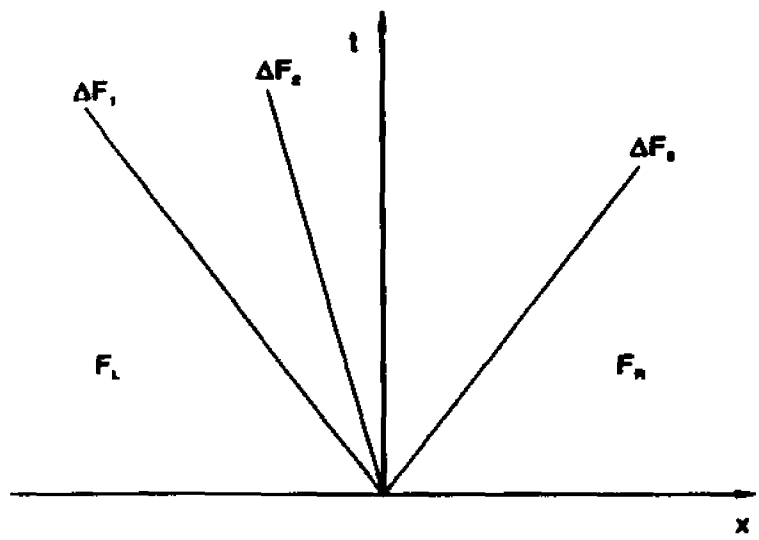


Figure 2.3: Evaluation of interface fluxes.

can be expressed as

$$\mathbf{F}_R - \mathbf{F}_L = (\mathbf{w}_R - \mathbf{w}_L) \sum_k \lambda_k$$

where λ_k is the speed of the kth wave emanating from the grid point at time t and each term in the summation represents the effect of a single wave. The flux at the cell interface can be computed by starting at either side of the interface and then summing the individual wave contributions up to the interface (See Fig. 2.3).

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}_L + (\mathbf{w}_R - \mathbf{w}_L) \sum_{k^-} \lambda_k$$

or

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}_R - (\mathbf{w}_R - \mathbf{w}_L) \sum_{k^+} \lambda_k$$

where the two summations above denote summation over all the negative and positive wave speeds, respectively. Taking the arithmetic average of these two expressions yields an expression for the interface flux.

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2} R(\mathrm{sgn}\Lambda) R^{-1}(\mathbf{w}_R - \mathbf{w}_L)$$

where R denotes the matrix of right eigenvectors of the Jacobian matrix D. $\mathrm{sgn}\Lambda$ is the diagonal matrix containing the signs of the eigenvalues of D and represents the characteristic direction of each wave. The eigenvalues of the Jacobian matrix D represent the characteristic speeds normal to the cell interface.

2.4.2 Steady Riemann Solvers

For the present work, Lawrence (1984), employs a sequence of two steady versions of the above-described one-dimensional Riemann problem to obtain the inviscid cross-flow numerical fluxes. These two steady Riemann problems can be expressed by the following generic expression

$$\frac{\partial \hat{\mathbf{E}}^*}{\partial \xi} = D_{m+\frac{1}{2}} \frac{\partial \hat{\mathbf{E}}^*}{\partial \kappa}, \tag{2.13}$$

with right and left states at opposite sides of the interface defined as

$$\hat{\mathbf{E}}^{*^n} = \begin{cases} \mathbf{E}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_m) & \text{for } \kappa < \kappa_{m+\frac{1}{2}} \\ \mathbf{E}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_{m+1}) & \text{for } \kappa > \kappa_{m+\frac{1}{2}} \end{cases}$$

where m is the index in the κ direction, $m + \frac{1}{2}$ denoting the location of the cell interface, and D is the constant Jacobian matrix defined by the generic expression

$$D_{m+\frac{1}{2}} = \left(\frac{\partial \hat{\mathbf{P}}_i}{\partial \hat{\mathbf{E}}^*}\right)_{m+\frac{1}{2}} = \left[\left(\frac{\kappa_x}{J}\right)\left(\frac{\partial \mathbf{E}_i}{\partial \hat{\mathbf{E}}^*}\right) + \left(\frac{\kappa_y}{J}\right)\left(\frac{\partial \mathbf{F}_i}{\partial \hat{\mathbf{E}}^*}\right) + \left(\frac{\kappa_z}{J}\right)\left(\frac{\partial \mathbf{G}_i}{\partial \hat{\mathbf{E}}^*}\right)\right]_{m+\frac{1}{2}} \tag{2.14}$$

The Riemann problem in the η-direction is obtained by replacing $\hat{\mathbf{P}}_i$ by $\hat{\mathbf{F}}_i$ and κ by η in Eq. (2.13). Similarly, the steady Riemann problem in the ζ-direction is obtained by replacing $\hat{\mathbf{P}}_i$ by $\hat{\mathbf{G}}_i$ and κ by ζ. In this manner, the one-dimensional Riemann problem is applied in each direction separately to obtain independent discretizations of the flux derivatives in each spatial direction. These independent results are added to obtain the overall discretization. It can be shown that if the flow properties which make up $D_{m+\frac{1}{2}}$ are carefully averaged between grid points at m and $m + 1$ so as to ensure that the following relation is satisfied, then

$$D_{m+\frac{1}{2}} \Delta(\hat{\mathbf{E}}^*)_{m+\frac{1}{2}} = \Delta(\hat{\mathbf{P}}_i)_{m+\frac{1}{2}}$$

When the flow is supersonic, a special averaging procedure [Roe, (1983)] insures that the above relation is satisfied. Cell interface values of density, velocity and enthalpy (which are used in the Jacobian matrix D) are computed as follows:

$$\begin{aligned} \rho_{m+\frac{1}{2}} &= \sqrt{\rho_m}\sqrt{\rho_{m+1}} \\ \phi_{m+\frac{1}{2}} &= \frac{\phi_{m+1}\sqrt{\rho_{m+1}} + \phi_m\sqrt{\rho_m}}{\sqrt{\rho_{m+1}} + \sqrt{\rho_m}} \end{aligned} \tag{2.15}$$

where $\phi = u$, v, w and $h = \gamma p/[(\gamma - 1)\rho] + \frac{1}{2}(u^2 + v^2 + w^2)$ is the enthalpy. From Eqs. (2.15), the sound speed, a, can be computed from

$$a_{m+\frac{1}{2}} = \sqrt{[h_{m+\frac{1}{2}} + \frac{1}{2}(u^2 + v^2 + w^2)](\gamma - 1)}$$

Once u, v, w and a are determined in this manner, the eigenvalues and corresponding left and right eigenvectors, which constitute the elements of the D matrix, are then constructed.

The solution of Eq. (2.13) for the inviscid cross-flow numerical flux at the cell interface is

$$\begin{aligned}(\hat{\mathbf{P}}_i^1)_{m+\frac{1}{2}} = &\left(\frac{\kappa_x}{J}\right)_{m+\frac{1}{2}} \frac{1}{2}\left[(\mathbf{E}_i)_m + (\mathbf{E}_i)_{m+1}\right] + \left(\frac{\kappa_y}{J}\right)_{m+\frac{1}{2}} \frac{1}{2}\left[(\mathbf{F}_i)_m + (\mathbf{F}_i)_{m+1}\right] \\ &+ \left(\frac{\kappa_z}{J}\right)_{m+\frac{1}{2}} \frac{1}{2}\left[(\mathbf{G}_i)_m + (\mathbf{G}_i)_{m+1}\right] \\ &- \frac{1}{2}(\mathrm{sgn}D)_{m+\frac{1}{2}}\left[\left(\frac{\kappa_x}{J}\right)_{m+\frac{1}{2}} \Delta\mathbf{E}_i + \left(\frac{\kappa_y}{J}\right)_{m+\frac{1}{2}} \Delta\mathbf{F}_i + \left(\frac{\kappa_z}{J}\right)_{m+\frac{1}{2}} \Delta\mathbf{G}_i\right]\end{aligned} \tag{2.16}$$

This expression consists of a central differencing component (first three terms) followed by a first order upwind dissipation term. The differencing operator Δ in the upwind term is the standard forward difference operator. Again using the terminology of Lawrence (1987), the matrix $\mathrm{sgn}D$, is defined as

$$\mathrm{sgn}D = R(\mathrm{sgn}\Lambda)R^{-1}$$

where R is the matrix whose rows consist of the eigenvectors of D (See Appendix A), and $\mathrm{sgn}\Lambda$ is the diagonal matrix with the ith element equal to -1 or 1 depending on the sign of the corresponding ith eigenvalue of D.

The streamwise location of the calculation of Eq. (2.16) is still unspecified. For the construction of an implicit algorithm, as will be further discussed in the next section, the fluxes of Eq. (2.16) are linearized about the $n+1$ marching station in the following manner

$$\begin{aligned}(\hat{\mathbf{F}}_i^1)^{n+1}_{k+\frac{1}{2},l} &= (\hat{\mathbf{F}}_i^1)^n_{k+\frac{1}{2},l} + \left[\frac{\partial(\hat{\mathbf{F}}_i^1)_{k+\frac{1}{2},l}}{\partial\mathbf{U}_{k+1,l}}\right]^n \delta\mathbf{U}_{k+1,l} + \left[\frac{\partial(\hat{\mathbf{F}}_i^1)_{k+\frac{1}{2},l}}{\partial\mathbf{U}_{k,l}}\right]^n \delta\mathbf{U}_{k,l} \\ (\hat{\mathbf{F}}_i^1)^{n+1}_{k-\frac{1}{2},l} &= (\hat{\mathbf{F}}_i^1)^n_{k-\frac{1}{2},l} + \left[\frac{\partial(\hat{\mathbf{F}}_i^1)_{k-\frac{1}{2},l}}{\partial\mathbf{U}_{k-1,l}}\right]^n \delta\mathbf{U}_{k-1,l} + \left[\frac{\partial(\hat{\mathbf{F}}_i^1)_{k-\frac{1}{2},l}}{\partial\mathbf{U}_{k,l}}\right]^n \delta\mathbf{U}_{k,l}\end{aligned} \tag{2.17}$$

$$(\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}^{n+1} = (\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}^{n} + \left[\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}}\right]^n \delta\mathbf{U}_{k,l+1} + \left[\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}\right]^n \delta\mathbf{U}_{k,l}$$

$$(\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}^{n+1} = (\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}^{n} + \left[\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}}\right]^n \delta\mathbf{U}_{k,l-1} + \left[\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}\right]^n \delta\mathbf{U}_{k,l} \tag{2.18}$$

The derivation of the inviscid Jacobians, $\partial\mathbf{F}_i/\partial\mathbf{U}$ and $\partial\mathbf{G}_i/\partial\mathbf{U}$, is provided in Appendix *B*.

2.5 Second-Order Upwind Scheme

In this section, the first order algorithm described above is extended to second order accuracy. This is done in a manner so as to preserve the shock capturing characteristics of the first order algorithm while improving the order of accuracy in less sharply varying regions.

It is important to keep in mind that the basic objective is to balance the sum of the fluxes through the individual grid cells. The major obstacle now being hurdled is to numerically represent these fluxes at the cell faces in such a manner as to preserve their wave-like nature.

The total numerical fluxes in the η- and ζ-directions are given in terms of the first order fluxes of Eq. (2.17) and (2.18) with the addition of a second order corrective term. The forms of these cross-flow fluxes are now presented in this section. Using the terminology of Chakravarthy (1987), intermediate parameters are first defined as

$$(\mathcal{A}_1)_{m+\frac{1}{2}} = R^{-1}_{m+\frac{1}{2}}[\hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_m) - \hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_{m-1})]$$

$$(\mathcal{A}_2)_{m+\frac{1}{2}} = R^{-1}_{m+\frac{1}{2}}[\hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_{m+1}) - \hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_m)]$$

$$(\mathcal{A}_3)_{m+\frac{1}{2}} = R^{-1}_{m+\frac{1}{2}}[\hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_{m+2}) - \hat{\mathbf{E}}^*(d\mathbf{S}^n_{m+\frac{1}{2}}, \mathbf{U}_{m+1})]$$

where $R^{-1}_{m+\frac{1}{2}}$ is the matrix of left eigenvectors evaluated at the $m+\frac{1}{2}$ cell interface. When $(\mathcal{A}_2)_{m+\frac{1}{2}}$ is multiplied by the associated right eigenvectors, the change in the flux vector $\hat{\mathbf{E}}^*$ across the corresponding wave of the approximate Riemann

problem at the $m+\frac{1}{2}$ cell face is obtained. The remaining two $\mathcal{A}$ quantities above are required for increasing the accuracy of the scheme.

Each element in the $\mathcal{A}$ vectors, denoted by α_1^i, α_2^i and α_3^i, is then limited relative to one another using a slope-limiting operator. The resulting new vectors, denoted by $\tilde{\mathcal{A}}_1$, $\tilde{\mathcal{A}}_2$, $\tilde{\mathcal{A}}_3$ and $\tilde{\mathcal{A}}_4$ are constructed to achieve non-oscillatory shock capturing. The elements of these new vectors are

$$\begin{aligned}
(\tilde{\alpha}_1^i)_{m+\frac{1}{2}} &= \text{minmod}\left[(\alpha_1^i)_{m+\frac{1}{2}}, \beta(\alpha_2^i)_{m+\frac{1}{2}}\right] \\
(\tilde{\alpha}_2^i)_{m+\frac{1}{2}} &= \text{minmod}\left[(\alpha_2^i)_{m+\frac{1}{2}}, \beta(\alpha_1^i)_{m+\frac{1}{2}}\right] \\
(\tilde{\alpha}_3^i)_{m+\frac{1}{2}} &= \text{minmod}\left[(\alpha_2^i)_{m+\frac{1}{2}}, \beta(\alpha_3^i)_{m+\frac{1}{2}}\right] \\
(\tilde{\alpha}_4^i)_{m+\frac{1}{2}} &= \text{minmod}\left[(\alpha_3^i)_{m+\frac{1}{2}}, \beta(\alpha_2^i)_{m+\frac{1}{2}}\right]
\end{aligned} \tag{2.19}$$

The flux limiting operator defined as

$$\text{minmod}(x, y) = \text{sign}(x)\max[0, \min\{|x|, y\text{sign}(x)\}]$$

reduces the accuracy of the scheme in the vicinity of flow-field discontinuities so that the oscillatory nature characteristic of second order methods is eliminated. Generally, in all regions of the flow except near maxima and minima (shock waves), the minmod operator will return the unlimited flux values (the major portion of the flow-field will contain eigenvalues possessing the same sign). Thus, the truncation error of the unlimited scheme provides a good indication of the overall truncation error of the limited scheme. In Eq. (2.19) above, β is a compression parameter and is a function of an accuracy parameter, ϕ, to be described later.

$$\beta = \frac{3-\phi}{1-\phi}$$

Using the flux limited vectors $\tilde{\mathcal{A}}_1$, $\tilde{\mathcal{A}}_2$, $\tilde{\mathcal{A}}_3$ and $\tilde{\mathcal{A}}_4$, the numerical flux changes

which are used to construct the second order inviscid flux are defined as follows

$$(d\mathbf{F}_1^+)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}}(\Lambda^+_{m+\frac{1}{2}})\tilde{A}_1$$
$$(d\mathbf{F}_2^+)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}}(\Lambda^+_{m+\frac{1}{2}})\tilde{A}_2$$
$$(d\mathbf{F}_1^-)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}}(\Lambda^-_{m+\frac{1}{2}})\tilde{A}_3$$
$$(d\mathbf{F}_2^-)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}}(\Lambda^-_{m+\frac{1}{2}})\tilde{A}_4$$

The diagonal matrices $\Lambda^{\pm}$ consist of elements which are the positive and negative projections of the eigenvalues at each cell face and are defined by

$$\lambda^{i\pm}_{m+\frac{1}{2}} = \frac{1}{2}(\lambda^i_{m+\frac{1}{2}} \pm |\lambda^i_{m+\frac{1}{2}}|) \tag{2.20}$$

Finally, the second order inviscid numerical flux is defined in terms of the first order inviscid numerical fluxes, $\hat{\mathbf{P}}^1_i$, of Eqs. (2.17) and (2.18), which are written here in terms of the generic expression of Eq. (2.16) and second order correction terms

$$\begin{aligned}(\hat{\mathbf{P}}^2_i)_{m+\frac{1}{2}} = (\hat{\mathbf{P}}^1_i)_{m+\frac{1}{2}} &+ \frac{1-\phi}{4}(d\mathbf{F}_1^+)_{m+\frac{1}{2}} + \frac{1+\phi}{4}(d\mathbf{F}_2^+)_{m+\frac{1}{2}} \\ &- \frac{1+\phi}{4}(d\mathbf{F}_1^-)_{m+\frac{1}{2}} - \frac{1-\phi}{4}(d\mathbf{F}_2^-)_{m+\frac{1}{2}}\end{aligned} \tag{2.21}$$

The fluxes in the η-direction are obtained by replacing the generic subscript, $m+\frac{1}{2}$, by $k+\frac{1}{2}, l$ and the flux in the ζ-direction is obtained by replacing $m+\frac{1}{2}$, with $k, l+\frac{1}{2}$. The results are of the form:

$$(\hat{\mathbf{F}}^2_i)_{k+\frac{1}{2},l} = (\hat{\mathbf{P}}^2_i)_{k+\frac{1}{2},l}$$
$$(\hat{\mathbf{G}}^2_i)_{k,l+\frac{1}{2}} = (\hat{\mathbf{P}}^2_i)_{k,l+\frac{1}{2}}$$

Roe's scheme is known to occasionally yield non-physical numerical results which include so-called expansion shocks. The reason for this lies in the manner in which Roe's scheme treats an expansion. In order to help Roe's Riemann solver prevent this non-physical behavior, a technique developed by Harten (1984) is used. This

technique consists of replacing the absolute-value operator of Eq. (2.20) with the conditional operator

$$\psi(z) = \begin{cases} |z| & \text{where } z \geq \epsilon \\ \frac{z^2+\epsilon^2}{2\epsilon} & \text{where } z < \epsilon \end{cases}$$

where ϵ is a small parameter.

2.6 Implicit Solution Procedure

The first order numerical fluxes of Eqs. (2.17) and (2.18) are evaluated at the $n+1$ marching station, and the second order correction terms of Eq. (2.22) are evaluated at the nth marching station. After substituting the linearized fluxes into Eq. (2.11), the result is a penta-diagonal system which is too large to solve efficiently. Therefore, the system is approximately factored into a sequence of two block tri-diagonal systems as follows.

$$\begin{aligned} &\left\{ \frac{\partial(\hat{\mathbf{F}}_i^1)^n_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}} - \frac{\partial(\hat{\mathbf{F}}_v)^n_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}} \right\} \mathbf{Q}_{k+1,l} \\ &+ \left\{ \hat{A}^{+^n}_{k,l} + \left[\frac{\partial(\hat{\mathbf{F}}_i^1)^n_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} - \frac{\partial(\hat{\mathbf{F}}_v)^n_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} \right] - \left[\frac{\partial(\hat{\mathbf{F}}_i^1)^n_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} - \frac{\partial(\hat{\mathbf{F}}_v)^n_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} \right] \right\} \mathbf{Q}_{k,l} \\ &- \left\{ \frac{\partial(\hat{\mathbf{F}}_i^1)^n_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}} - \frac{\partial(\hat{\mathbf{F}}_v)^n_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}} \right\} \mathbf{Q}_{k-1,l} = RHS^n \end{aligned}$$

where the right-hand side is given by

$$\begin{aligned} RHS^n = -(\hat{A}^{+^n}_{k,l} - \hat{A}^{+^{n-1}}_{k,l})\mathbf{U}^n_{k,l} &- \left[(\hat{\mathbf{F}}_i^2 - \hat{\mathbf{F}}_v)_{k+\frac{1}{2},l} - (\hat{\mathbf{F}}_i^2 - \hat{\mathbf{F}}_v)_{k-\frac{1}{2},l} \right]^n \\ &- \left[(\hat{\mathbf{G}}_i^2 - \hat{\mathbf{G}}_v)_{k,l+\frac{1}{2}} - (\hat{\mathbf{G}}_i^2 - \hat{\mathbf{G}}_v)_{k,l-\frac{1}{2}} \right]^n \\ &- \left[\hat{\mathbf{E}}^p(d\mathbf{S}^{n+1}_{k,l}, \hat{\mathbf{U}}^n_{k,l}) - \hat{\mathbf{E}}^p(d\mathbf{S}^n_{k,l}, \hat{\mathbf{U}}^{n-1}_{k,l}) \right] \end{aligned}$$

The following is then solved to obtain the solution at the $n+1$ marching station.

$$\begin{aligned} &\left\{ \frac{\partial(\hat{\mathbf{G}}_i^1)^n_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}} - \frac{\partial(\hat{\mathbf{G}}_v)^n_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}} \right\} \delta\mathbf{U}_{k,l+1} \\ &+ \left\{ \hat{A}^{+^n}_{k,l} + \left[\frac{\partial(\hat{\mathbf{G}}_i^1)^n_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} - \frac{\partial(\hat{\mathbf{G}}_v)^n_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} \right] - \left[\frac{\partial(\hat{\mathbf{G}}_i^1)^n_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} - \frac{\partial(\hat{\mathbf{G}}_v)^n_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} \right] \right\} \delta\mathbf{U}_{k,l} \\ &- \left\{ \frac{\partial(\hat{\mathbf{G}}_i^1)^n_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}} - \frac{\partial(\hat{\mathbf{G}}_v)^n_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}} \right\} \delta\mathbf{U}_{k,l-1} = \hat{A}^{+^n}_{k,l} \mathbf{Q}_{k,l} \end{aligned}$$

The viscous flux vectors and Jacobians, $\hat{\mathbf{F}}_v$ and $\hat{\mathbf{G}}_v$, and $\partial\mathbf{F}_v/\partial\mathbf{U}$ and $\partial\mathbf{G}_v/\partial\mathbf{U}$, are provided in Appendix C and are a result of a linearization of the same form as in Eqs. (2.17) and (2.18).

The truncation error of the unlimited scheme, which was earlier argued to be a good approximation of the truncation error for the limited scheme, can be written as

$$\mathrm{TE} = -\frac{1}{4}\left(\phi - \frac{1}{3}\right)(\Delta\kappa)^2\left[\frac{\kappa_x}{J}\frac{\partial^3\mathbf{E}_i}{\partial\kappa^3} + \frac{\kappa_y}{J}\frac{\partial^3\mathbf{F}_i}{\partial\kappa^3} + \frac{\kappa_z}{J}\frac{\partial^3\mathbf{G}_i}{\partial\kappa^3}\right]$$

Thus, as mentioned at the beginning of this section, schemes of varying accuracy can be obtained simply by altering the value of the accuracy parameter, ϕ. Chakravarthy and Osher (1985) provide a table illustrating the various schemes obtainable by changing the value of this parameter. For example, a value of $\phi = 1/3$, produces a third order accurate scheme.

2.7 Grid Generation

There are many reported studies on the subject of numerical grid generation for three-dimensional applications [Thompson et al., (1985)]. The purpose of this section is to provide a brief description of the grid generation methods used for the present work.

The present algorithm incorporates several methods for obtaining a numerical grid for discretization of the governing equations. For simple conical geometries, an algebraic grid generation technique is used. This method, which is considerably faster than solving partial differential equations, was chosen for all of the test cases to appear in Chapters 4 and 6. The algebraic grid generator, which generates grid planes perpendicular to the x-coordinate, begins by distributing points evenly in the circumferential direction along the body surface (the body surface must be analytically defined). Constant η-lines are then defined to be straight line segments radiating outward from the body and extending to an analytically

determined elliptically shaped outer grid line. Finally, grid points are clustered along the constant η-lines according to the Roberts exponential stretching function. See, for example (Anderson, Tannehill and Pletcher, 1984).

$$z(\xi,\eta,\zeta) = \frac{\beta(\xi,\eta) + 1 - [\beta(\xi,\eta) - 1]\left[\frac{\beta(\xi,\eta)+1}{\beta(\xi,\eta)-1}\right]^{1-\eta}}{\left[\frac{\beta(\xi,\eta)+1}{\beta(\xi,\eta)-1}\right]^{1-\eta} + 1}$$

Here, $\beta(\xi,\eta)$ is found using an iterative procedure to give a user requested wall spacing. The position vectors of the grid points are then given by

$$\mathbf{r}(\xi,\eta,\zeta) = \mathbf{r}(\xi,\eta,0) + z(\xi,\eta,\zeta)[\mathbf{r}(\xi,\eta,\zeta_{max}) - \mathbf{r}(\xi,\eta,0)]$$

where $\zeta = 0$ at the body surface and $\zeta = \zeta_{max}$ at the outer edge of the grid. First, a grid plane is generated in this manner at the current x-location. Another plane is then created at a pre-determined distance downstream. Intermediate marching stations are obtained between these two x-stations by linear interpolation.

If the body cannot be defined analytically, a hyperbolic grid generator exists within the code which distributes grid points according to the solution of a set of hyperbolic partial differential equations. An externally-generated surface grid is first read from a grid file. The grid is then stretched outward from the body to a pre-determined outer grid line.

For more complex body configurations, a preliminary grid must be generated by some external numerical grid generation technique. (See, for example, Thompson, Warsi and Mastin, 1985). The present computer code has a mechanism for reading a binary grid file one ξ-plane at a time. After the flow solution has been computed at each ξ-location, the code checks to see if the next ξ-location is within the bounds of the two pre-existing grid planes. If so, a linear interpolation is performed to obtain a new grid plane. Otherwise, a new plane is read from the grid file (or a new plane is created by the algebraic routine).

Later, in the sections to follow, a method of determining the extent by which the cross-flow grid at each marching station should extend out into the freestream is presented. This farfield extent of the grid is obtained by monitoring the freestream conditions and maintaining a minimum number of grid points in the freestream. This procedure not only maximizes grid point density inside the region of interest but also provides a farfield grid boundary at each marching station. As we will see in later sections, this information can be used to construct a grid plane at the next marching station by projecting the grid at the current marching station downstream using the surface grid as a guide. A more detailed discussion of this is given after the grid adaptation procedure has been described.

CHAPTER 3

Two-Dimensional Adaptive Grid Procedure

In this chapter a two-dimensional solution-adaptive grid algorithm is developed for use in the flow algorithm described in the previous section. The method for selecting an error estimate is also contained in this chapter. A grid-fitting scheme is introduced for two-dimensions which monitors the total extent of the region of interest and maintains a maximum grid point density in this region. The resulting algorithm proceeds by first obtaining a preliminary solution at the current marching station. Based on this preliminary solution, a solution error estimate is computed at each grid point, and the grid spacing is adjusted so as to minimize this error. An improved solution is then obtained on this newly refined grid. The adapted grid is then projected downstream to the next marching station in preparation for the calculation at the next step.

3.1 Adaptation Strategy

Since the algorithm described in the previous section marches in space rather than time, each streamwise location of the flow-field is solved with the upstream condition as the corresponding initial condition. An appropriate adaptive grid scheme would construct a preliminary cross-flow grid plane at the streamwise location of interest, solve for the flow-field variables at this location based on the upstream conditions and then re-cluster the grid points based on the gradients and/or curvature of the computed flow. An improved solution can then be obtained on the resulting adapted cross-flow grid plane. A new preliminary cross-flow grid plane can then be constructed at the next streamwise location (with grid points proportioned to that of the previous plane) and the procedure continued.

The basic philosophy adopted in the present work to re-cluster grid points is somewhat similar to that of Gnoffo (1982) and is analogous to minimizing the energy in a system of tension springs set between grid points. The spring constants represent the error estimate and are functions of the gradient and/or curvature of a selected dependent variable. Nakahashi and Deiwert (1984) later refined Gnoffo's method by introducing the concept of a torsion spring to reduce grid skewness between grid lines or planes. Although the method is based on variational principles, the problem is posed by Nakahashi and Deiwert in an algebraic uni-directional manner by applying tension and torsion spring forces proportional to flow gradients at each grid point along a fixed coordinate line and solving for the equilibrium position of the resulting grid system. The resulting system of equations is tri-diagonal, which can be efficiently solved for the final positions of the grid points.

The present grid adaptation technique is based on the error equi-distribution method and involves the re-distribution of grid points such that an error measure,

which is generally proportional to $(\Delta s)^n$ multiplied by a function of the flow-field variable derivatives (i.e. $\Delta s^n \omega(\phi, \phi', \cdots, \phi^n)$), is equally distributed over a coordinate line. In simplistic terms, this can be written as

$$w_i \Delta s_i = K \tag{3.1}$$

where w_i, the weighting function, is based on flow properties, and in the terminology of [Nakahashi and Deiwert (1985)] and [Davies (1988)], represents the spring constant with K as the resultant force. The grid interval, Δs_i, is defined as the distance between adjacent grid points along a line of constant computational coordinate. Figure 3.1 illustrates the grid topology surrounding the grid point (i, j).

The weighting function, w_i, is defined as a function of a selected normalized flow property, $\bar{f}$ such that

$$w_i = 1 + A \bar{f}_i^B \tag{3.2}$$

where A and B are constants related to the specified maximum and minimum grid spacings. From Eq. (3.1), the maximum and minimum allowable grid spacing occur when w_i is at its minimum and maximum values, respectively. Since the normalized flow property, $\bar{f}$, varies from zero to unity, Eq. (3.2) states that $w_{max} = 1 + A$ and $w_{min} = 1$. Upon substitution of these values into Eq. (3.1) and rearranging yields $A = \Delta s_{MAX} / \Delta s_{MIN} - 1$, where Δs_{MAX} and Δs_{MIN} are user specified maximum and minimum allowable grid spacings. The constant B of Eq. (3.2) is chosen so that the computed minimum grid spacing corresponds to Δs_{MIN}, the requested minimum grid spacing. Davies et al. (1989) outlines an iterative procedure for obtaining an approximate value for B. For completeness, this procedure is included in Appendix D. The choice of the normalized flow property $\bar{f}$ is described in a section to follow.

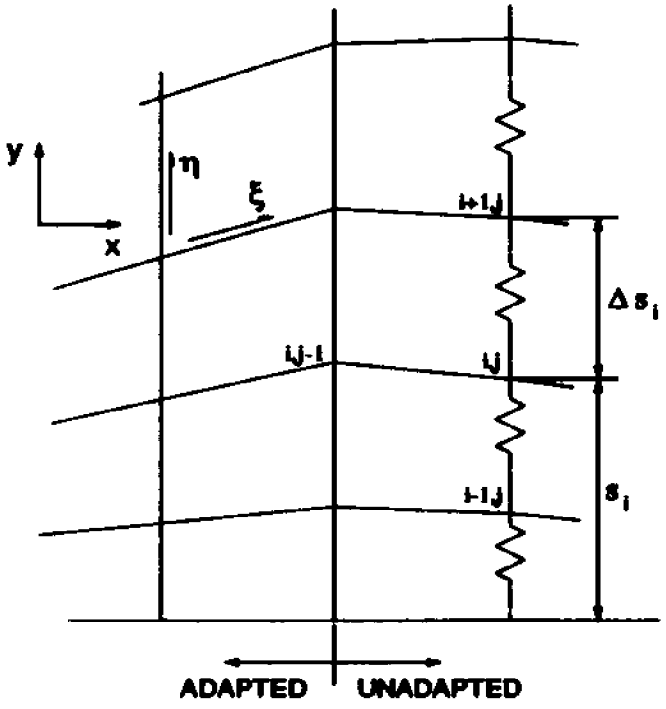


Figure 3.1: Adaptive mesh geometry.

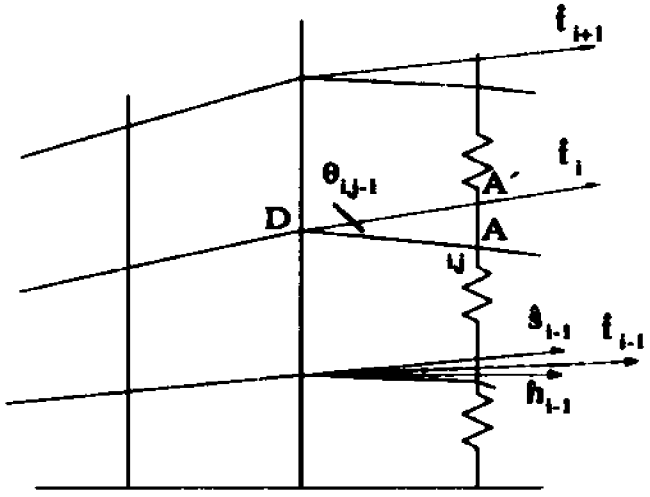


Figure 3.2: Tensional and torsional spring system.

Summing both sides of Eq. (3.1), solving for K, and then substituting the resulting expression for K back into Eq. (3.1) yields

$$w_i \Delta s_i = s_{max} / \left(\sum_{l=1}^{n} \frac{1}{w_l} \right) \tag{3.3}$$

where s_{max} is the total length of the coordinate line. Equation (3.3) is an expression for the new grid point locations based on local flow properties. Thus, where the error measure, w_i, is large, Δs_i will turn out to be small and, therefore, the local truncation error at a point, whose leading term is proportional to $w_i(\Delta s_i)^n$, will tend to be equi-distributed over the complete domain.

Using Eq. (3.3), a tri-diagonal system of equations can easily be set up for the grid point positions. For example, applying this equation at the ith point leads to

$$w_i(s_{i+1} - s_i) - w_{i-1}(s_i - s_{i-1}) = 0$$

However, if this equation is used alone to determine the new grid spacing along each grid line, the grid would quickly become skewed due to the lack of dependence of the adapted grid points on the previous grid line. To ensure a smoother grid there needs to be a relationship between the current grid adaptation line (where the grid adaptation is to be performed) and the previously adapted grid line. Torsional terms are introduced to accomplish this task. Figure 3.2 graphically illustrates the addition of a torsion spring to the system. This procedure, developed by Nakahashi and Deiwert, relates grid points along adjacent grid lines thus providing for a smoother grid. Performing a force balance on the system of tension and torsion springs acting on the point (i, j) of Fig. 3.2, the following equation is obtained.

$$w_i(s_{i+1} - s_i) - w_{i-1}(s_i - s_{i-1}) - C_i\theta_{i,j-1} = 0$$

where $-C_i\theta_{i,j-1}$ denotes the force due to the torsion spring. The torsion angle, θ, can be approximated by $\theta_i \approx (s_i - s'_{s,i})/|DA'|$, where, $|DA'|$ is the length of the

line segment from the point $(i, j-1)$ to the point of intersection of the torsional vector, $\hat{\mathbf{t}}_i$, with the current adaptation line. The quantity $s'_{s,i}$ is the physical distance to the point where the streamwise torsional reference line crosses the current adaptation line and corresponds to the point A' of Fig. 3.2. Substituting this expression for the last term in the above equation and rearranging yields

$$w_{i-1}s_{i-1} - (w_{i-1} + w_i + \tau_{s,i})s_i + w_i s_{i+1} = -\tau_{s,i}s'_{s,i} \tag{3.4}$$

which is tri-diagonal, and can be easily solved for s_i. The constant, $\tau_{s,i} = C_i/|DA'|$, is the magnitude of the torsional influence from the streamwise direction. The choice of C was observed to greatly effect consistency in grid smoothness and continuity. Nakahashi and Deiwert (1986) set C proportional to the average of the tension spring constants along a line. Davies and Venkatapathy (1989) assumed that C is proportional to the maximum w_i and the local aspect ratio of the grid cell. This method is adopted in the present study with the modification that C is also assumed proportional to the length of the torsion spring, $|DA'|$. As a result, $\tau_{s,i}$ is expressed in the following manner.

$$\tau_{s,i} = \frac{\lambda w_{max}(s_{i+1,j-1} - s_{i-1,j-1})}{2|DA'|}$$

The constant, λ, is used as a user input to control the amount of torsion or 'rigidity' the system possesses and is usually on the order of 10^{-1} to 10^{-4}. A value of zero for λ completely removes torsional dependence and allows grid points to move without influence from adjacent grid lines. The torsional reference line, DA', is constructed in the direction of the torsional vector, $\hat{\mathbf{t}}_i$, which, in the present work is taken to be proportional to the orthogonal and straightness vectors (See Fig. 3.2). The orthogonal vector, $\hat{\mathbf{n}}_i$, is defined as a vector in the positive x-direction and represents the direction which the grid must take in order to maintain a modest degree of overall orthogonality. The straightness vector, $\hat{\mathbf{s}}_i$,

at the point (i,j) is the average of two vectors. The first of these vectors is established from the point $(i,j-2)$ to the point $(i,j-1)$ and the second is taken from the point $(i,j-1)$ to the point (i,j). The torsional vector, $\hat{\mathbf{t}}_i$, is defined as

$$\hat{\mathbf{t}}_i = C_t\hat{\mathbf{s}}_i + (1 - C_t)\hat{\mathbf{n}}_i$$

where C_t is a user input. A value of C_t close to zero leads to a more orthogonal grid while C_t closer to unity adds more straightness to the grid. The torsional reference line, DA' acts in the direction of the torsional vector, $\hat{\mathbf{t}}_i$.

As noted earlier, the constants A and B used to compute the weighting functions defined by Eq. (3.2) were derived with the 'torsionless' equations (Eq. 3.3). With the addition of the torsion terms, the computed minimum and maximum grid spacing will be somewhat altered and may not correspond to the requested Δs_{MIN} and Δs_{MAX}. To help alleviate this problem and provide for improved convergence, a weighting of the form

$$w_i = (1 + A\bar{f}_i^B)w_{t_i}$$

is used instead of Eq. (3.2). When Eq. (3.4) is solved, each Δs_i is checked to see if it lies within the user requested limits of Δs_{MIN} and Δs_{MAX}. If so, the value of w_{t_i} is set equal to unity. If Δs_i is not within these limits, the conditional operator

$$w_{t_i} = \begin{cases} \frac{1}{2}\left(\frac{\Delta s_i}{\Delta s_{MIN}} + 1\right) & \text{for } \Delta s_i < \Delta s_{MIN} \\ \frac{1}{2}\left(\frac{\Delta s_i}{\Delta s_{MAX}} + 1\right) & \text{for } \Delta s_i > \Delta s_{MAX} \end{cases} \tag{3.5}$$

is used to define w_{t_i}. Although Eq. (3.5) does not guarantee that Δs_i will lie within the user requested maximum and minimum grid spacings, its use is observed to enhance the stability of the algorithm.

Special treatment of weighting functions may be required at the boundaries of the adaptation domain, which can include all or be a subset of the entire calculation domain. This task is accomplished by adjusting the edge weighting function

to correspond with the desired edge spacing at each iteration. If boundary spacing exceeds a specified boundary spacing, Δs_{edge}, by a prescribed tolerance, the weighting function at the edge is then adjusted in the following manner.

$$w_{edge} = \frac{P}{\Delta s_{edge}}$$

where P is the average product, $w_i \times \Delta s_i$, over the entire line, which is assumed constant. The surrounding weighting functions are then smoothed to provide continuous spacing outward from the boundary of the adaptation domain.

3.2 Selection of Weighting Functions

The selection of weighting functions and the normalized flow variable, $\bar{f}$, used to drive the grid adaptation is considered next. Ideally, it is desired to minimize the truncation error and distribute the error uniformly over the computational domain. As noted before, although the truncation error cannot be estimated accurately, it is generally proportional to $(\Delta s)^n \partial^n \phi / \partial s^n$, where n denotes the order of accuracy, and for simplicity in the present work, it is set equal to $w\Delta s = (1 + A\bar{f})^B \Delta s$, where $\bar{f}$ is the normalized weighting function.

As in many of the studies on solution-adaptive gridding, a linear combination of gradients and curvature of selected dependent and/or physical variables is used here as weighting functions. Finding a flow-field variable which will consistently represent solution error in all regions of the solution domain is difficult, if not impossible. Some variables change very rapidly in certain regions, but remain fairly smooth in other regions where another variable could be undergoing severe change. In the present study, an algorithm is developed that automatically chooses which variables are to represent the weighting function at each grid point location. The scheme computes gradients and/or the curvature of each user specified variable and then, after normalizing each of these, chooses the largest in magnitude to represent the weighting function at each point. This method has

proven superior to conventional weighting function selection processes, especially where large changes in different dependent variables exist in different regions of the flow.

The normalized variable, $\bar{f}_i$, at a cross-stream location i and at the current streamwise location is computed as

$$\bar{f}_i = \frac{f_i - f_{min}}{f_{max} - f_{min}}$$

where f_i is a linear combination of the gradients and curvature of the dependent flow variables, ϕ_k.

$$f_i = \alpha\left(\frac{\partial \phi_i}{\partial s}\right) + \beta\left(\frac{\partial^2 \phi_i}{\partial s^2}\right) \tag{3.6}$$

where

$$\frac{\partial \phi_i}{\partial s} = \max_k \left|\frac{\partial \phi_i}{\partial s}\right|_k, \qquad \frac{\partial^2 \phi_i}{\partial s^2} = \max_k \left|\frac{\partial^2 \phi_i}{\partial s^2}\right|_k$$

for all specified ϕ_k's, (i.e., Mach number, pressure, density etc.). Hence, the gradients of all selected flow variables are computed, and then the maximum is chosen to represent f at each point along the current adaptation line. A similar procedure is done for the curvatures if β is not equal to zero. The constants α and β are specified by the user. Later, in subsection 4.4, after solving a few problems, we will return to the subject of weighting functions. The effects of gradient and curvature will be studied as applied to an actual flow problem. The weighting function selection technique of this section will also be compared with other weighting function selection methods.

The solution-grid adaptation procedure is done in a two-step manner. Once the adaptive grid solution has been obtained at a certain streamwise location, the first step in obtaining the solution at the next streamwise location is to project the just-obtained adapted grid downstream. An initial solution is then obtained on this grid. Based on this solution the new grid is adapted and, in the second step,

a refined solution is re-computed. Once this is done, the calculation proceeds to the next streamwise location, and the two-step procedure is repeated.

3.3 Grid-Fitting Algorithm

The present flow solver incorporates a shock capturing solution procedure for high Reynolds number external flow situations. At each marching station, to successfully implement farfield boundary conditions, a grid plane must first be constructed which extends a sufficient distance into the freestream. It would obviously be desirable to minimize the number of grid points placed in the freestream, keeping a maximum number of points inside the important region of the flow-field.

Shock-capturing methods are the most widely used techniques for solving the inviscid equations containing shock waves. In this approach, the Euler equations are cast in strong conservation form and any discontinuities are predicted as part of the solution with no special treatment required. Although the shock waves predicted by shock-capturing methods are smeared over several mesh points, their simplicity usually outweighs the slight compromise in results compared with the added complexities involved with shock-fitting schemes. Another advantage of shock-capturing schemes over shock-fitting arises when complicated internal shock structures exist. In this case, the internal structure is predicted without the special treatment of each shock wave required by shock-fitting schemes. With shock-fitting methods the shock is fitted as a boundary of the solution domain, and its position is computed as part of the solution. The process of shock-fitting is a matter of satisfying the Rankine-Hugoniot equations while simultaneously requiring that the solution on the downstream side of the shock be compatible with the rest of the flow-field.

Shock-capturing versus shock-fitting can be thought of as the technique employed to treat the farfield boundary of the flow. In shock-capturing the solution domain is extended a sufficient distance away from the solid boundary that

freestream conditions are assured and can be applied to the farfield. With shock-fitting schemes the farfield or outer boundary of the flow is coincident with the outermost shock wave. The Rankine-Hugoniot (R-H) relations can be applied across this boundary with freestream conditions as the upstream conditions. As noted before, the flow-field conditions downstream of the shock must be kept compatible with downstream shock conditions dictated by the R-H relations. A result of the shock-fitting method is a shock wave that is indeed discontinuous at a mesh interval. Another advantage of the shock-capturing method over the shock-fitting technique is that no prior knowledge of initial shock position is required; however, extra points must be added to the flow-field to insure that the farfield boundary completely encompasses the shock. These points are wasted insofar as useful flow-field information is concerned, existing only for the purpose of applying the freestream boundary conditions.

The grid-fitting scheme introduced in this section incorporates the reduced calculation domain inherent in shock-fitting schemes with the relatively simple far-field boundary condition of the shock-capturing method. The present grid-fitting scheme locates the innermost bound of the freestream conditions at the current marching station by scanning outward along the marching station, comparing flow-field conditions at each point to freestream conditions. The outermost grid line is then adjusted to a prescribed distance above this newly defined freestream boundary, minimizing the cross-stream calculation domain. The interior grid points are then adapted inside this reduced domain.

This procedure can be summarized as follows. A preliminary solution is first obtained on an initial grid at the current marching station. A search of flow-field variables is then performed outward from the solid boundary to find the point where freestream conditions are first encountered. The outermost grid point is then placed a small distance above this position (which is a prescribed percent

of the total length of the grid line). It is in this manner that the geometrical boundary of the solution domain is altered during the course of the calculations, reducing the number of unnecessary grid points present in the freestream and moving them to more important regions of the flow-field. Next, the interior grid points are re-distributed inside this reduced domain using the adaptive technique described in previous sections. A new, improved solution is then computed on the refined mesh. The refined mesh (including the total height) is then projected downstream to the next marching station location and the procedure is repeated. At each marching station the outermost grid boundary is monitored, maintaining a minimum number of grid points outside the shock wave.

Another advantage of the grid-fitting procedure is that the resulting computational grid becomes accurately aligned with the flow-field. As the present grid adaptation algorithm evolves (in succeeding chapters) for application to complex three-dimensional geometries, this additional advantage of the grid fitting concept will be further illustrated.

CHAPTER 4

Two-Dimensional Applications

A variety of applications are presented which illustrates the utility of the adaptive grid methods presented in the previous section. These applications include the hypersonic flow over a 15° compression surface, the hypersonic flow through a cowl-type inlet configuration and the inviscid axisymmetric flow over a cone-cylinder geometry. The first two applications represent problems of general interest in aerospace while the last problem is of interest in sonic boom predictions.

4.1 Hypersonic Compression Corner

The first test case involves hypersonic laminar flow over a 15° compression corner. Lawrence et al. (1984) have computed numerical results for this case. This case has also been studied experimentally by Holden and Moselle (1969), and the resulting pressure and heat transfer measurements are used here for comparison with present results. Figure 4.1 illustrates the basic geometry and flow-field structure. The 15° compression corner is sufficient to produce a strong shock wave which interacts with the leading edge shock to produce an interaction of the type IV as classified by Edney (1968). The flow remains attached at the base of the ramp, thus making it an excellent simple case for demonstrating the utility of the present solution-adaptive parabolized Navier-Stokes algorithm.

The flow conditions are given below.

$$M_\infty = 14.1 \qquad \tilde{T}_\infty = 72.2K \qquad \gamma = 1.4$$

$$Re_l = 1.04 \times 10^5 \qquad \tilde{T}_w = 297K \qquad Pr = 0.72$$

where Re_l is the freestream Reynolds number based on the distance, $\tilde{l} = .439m$, from the leading edge to the beginning of the ramp. The flow-field contains multiple shock waves which require a high degree of grid resolution to be captured clearly. Mach number and pressure were selected as the ϕ_k's of Eq. (3.6) ($k = 2$), and α and β were set equal to 1.0 and 0.1, respectively. An initial plane was generated at $x = 0.015m$ by exponentially stretching 45 grid points in the normal direction. The solution process was started at this location from freestream conditions using a step size of $10^{-4}m$. Grid adaptation was initiated at $x = 0.05m$ where the marching step size was increased to $10^{-3}m$. The maximum and minimum allowable grid spacings, Δs_{MAX} and Δs_{MIN}, were chosen to be 4.5 and 0.3 times the average upstream grid spacing, respectively. The torsional parameter, λ, was set at $5 \times 10,^{-5}$ and the orthogonality parameter, C_t, was set to $\frac{1}{2}$. Figure 4.2 shows the geometry and the resulting grids. For clarity, in each part of the

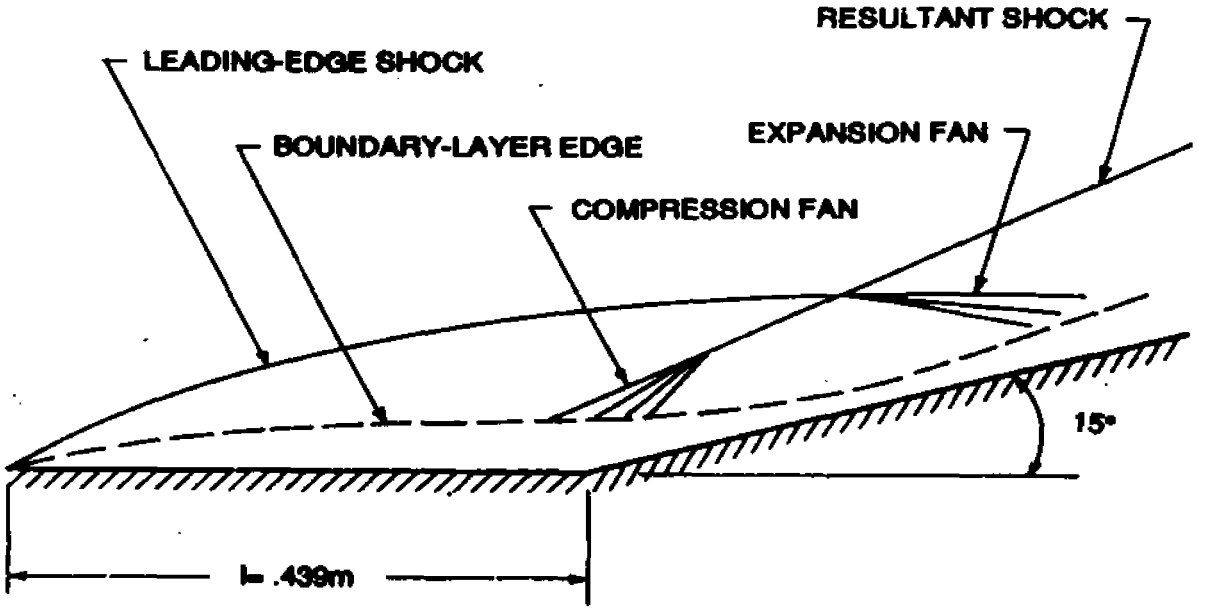


Figure 4.1: Compression corner geometry.

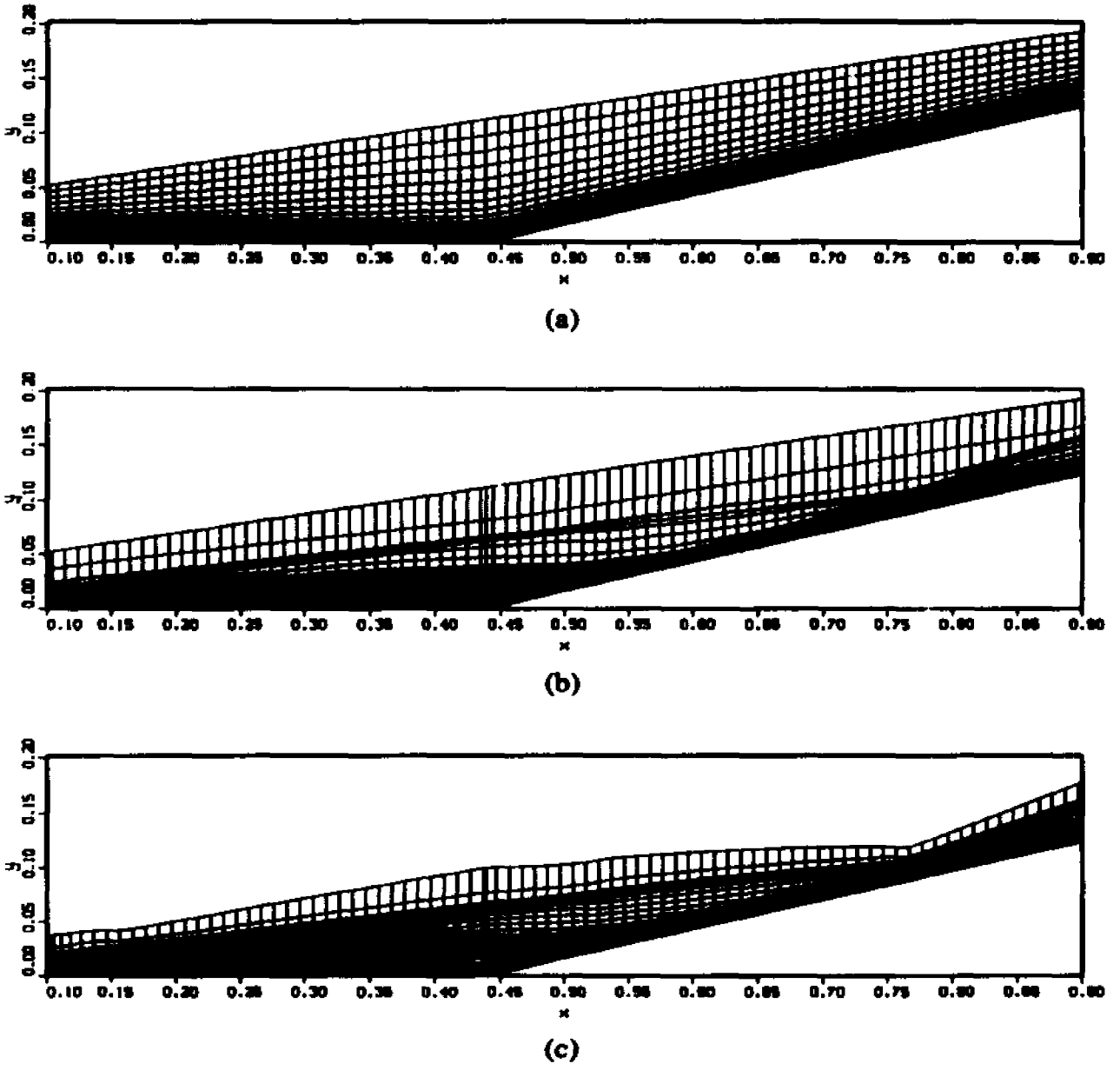


Figure 4.2: Geometry and computational grids for hypersonic flow over a compression corner for three test cases. (a) Without solution adaptive gridding, (b) with solution adaptive gridding and, (c) with adaptive gridding and grid-fitting. Every second grid line is shown in the normal direction for all three grids. All dimension are in meters.

figure, every other grid line is printed in the normal direction while every tenth grid line is shown in the streamwise direction. Figure 4.2*a* shows the grid structure without adaptive gridding. In Fig. 4.2*b* grid adaptation is performed, but without any grid-fitting. The outermost grid line of Fig. 4.2*c* was positioned using the grid-fitting algorithm described earlier. Gradients of density and streamwise velocity component were used to locate the innermost bound for the freestream conditions.

Figure 4.3 illustrates the region surrounding the leading-edge-ramp shock interaction. All normal lines are shown. For the adapted cases, point density has increased in the appropriate regions to resolve the shock waves. The grid-fitted adapted case shows increased grid point density in virtually all regions of the flow. For this case increased grid clustering is evident around the expansion fan emanating from the point of intersection of the two shock waves. In both adapted cases, wall spacing was allowed to decrease if necessary, but not allowed to increase beyond $10^{-4}m$.

Figure 4.4 shows contours of constant pressure in the region of interest around the shock wave interaction. The adapted case shows increased shock wave resolution over the unadapted case, even over the unadapted case employing twice as many grid points. The pressure contour oscillations evident in the ramp shock region of Fig. 4.4*a* are not present in Figs. 4.4*b* and 4.4*d* due to the ability of the grid adaptation scheme to identify these high gradient regions and locally increase grid point density. Another reason for the absence of this waviness in the adapted cases is the improved alignment of the grid with the shock waves. This alignment of the shocks with the computational grid has a stabilizing effect on the PNS solver. The pressure contours of Fig. 4.4 clearly illustrate the intersection of the bow shock with the ramp shock. These features are less clearly resolved for the unadapted cases. The figure corresponding to the grid-fitted adapted mesh

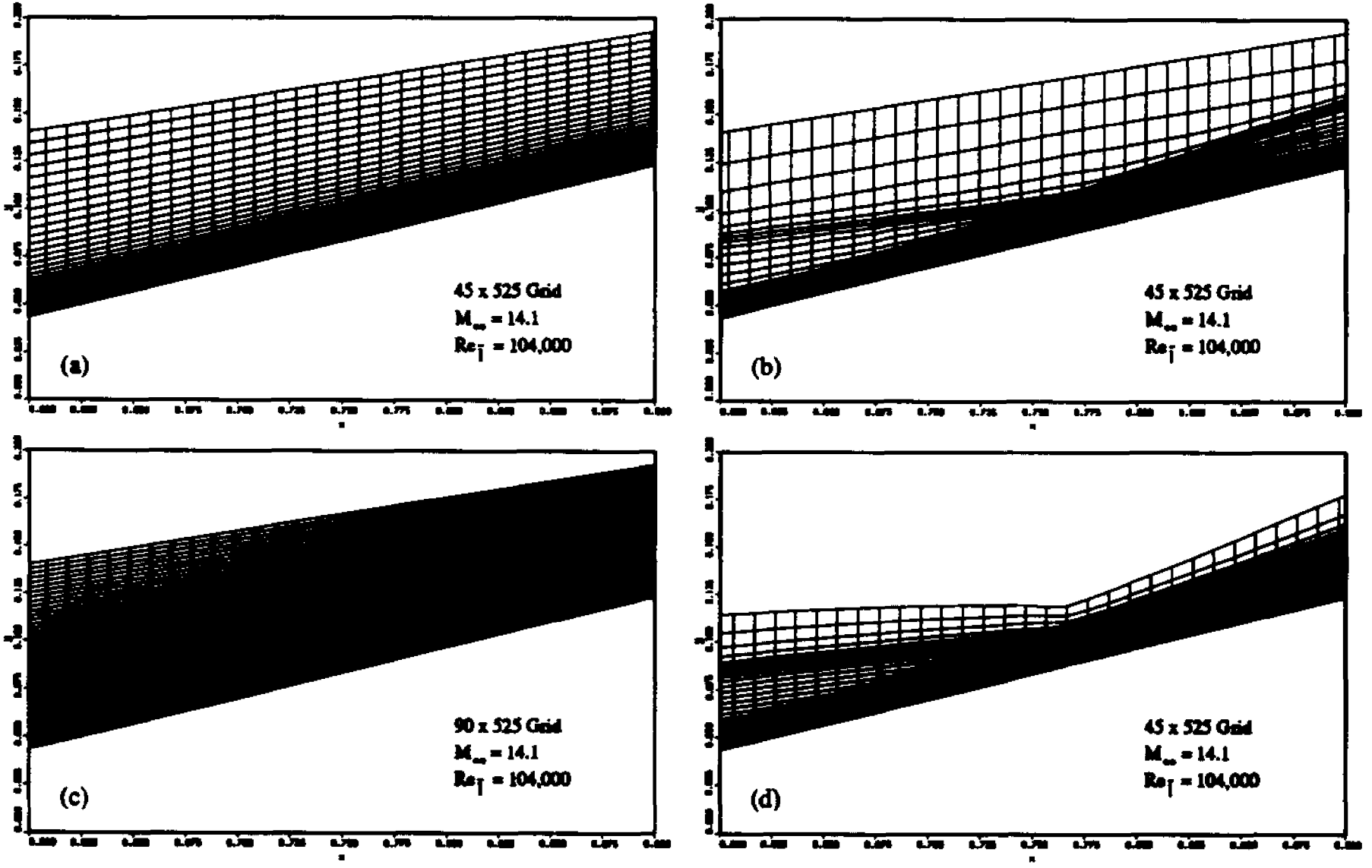


Figure 4.3: Computational grids in the vicinity of the shock interaction region. (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting. All dimensions are shown in meters.

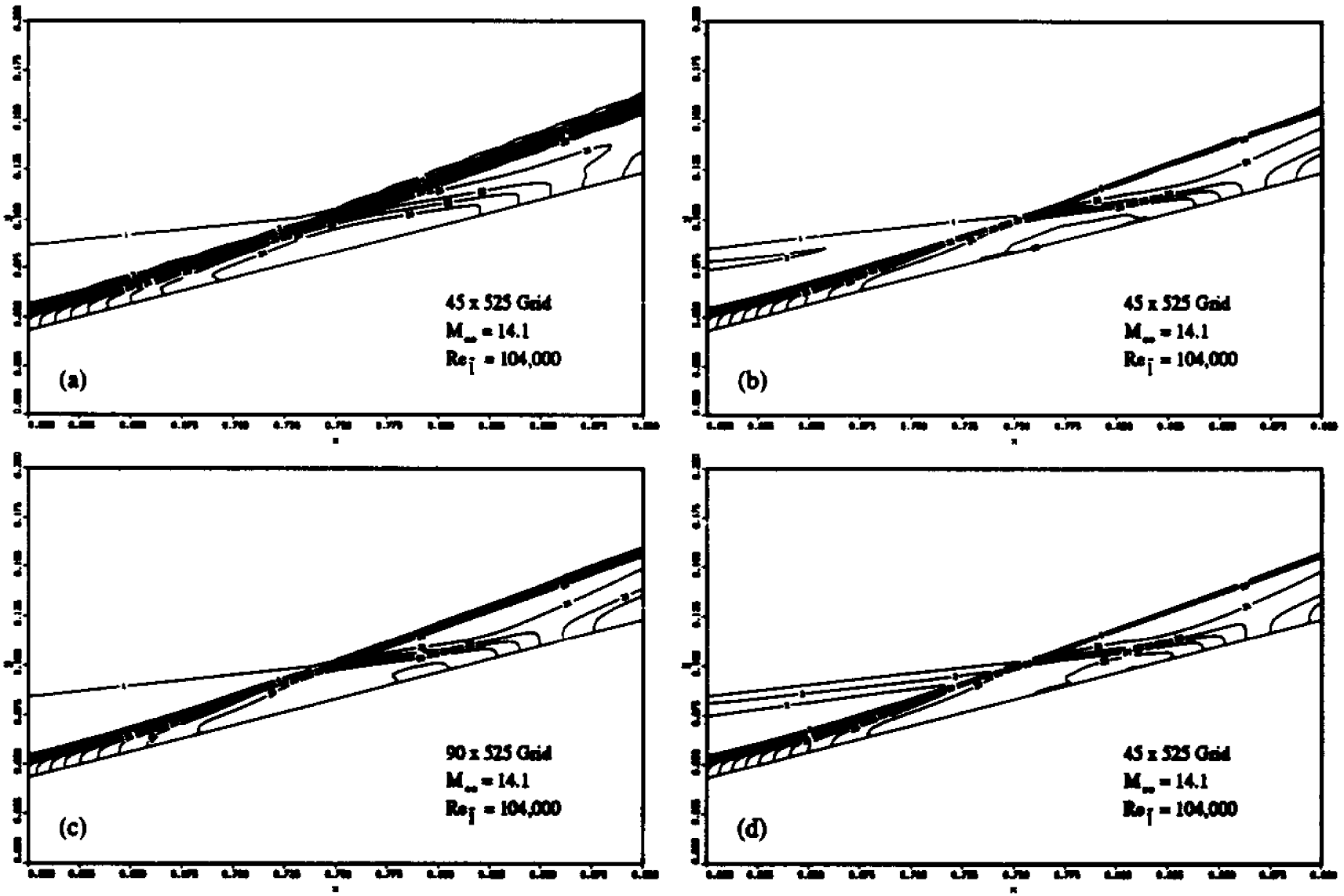


Figure 4.4: Pressure contours in the shock interaction region. (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting.

algorithm developed in the present work, appears to be optimal in terms of clearly resolving shock waves.

Contours of constant Mach number for each of the four cases are shown in Fig. 4.5, further illustrating the improvements obtainable with the solution-adaptive algorithm. In addition to the shock locations, the contact surface emanating from the intersection point of the shock waves is more clearly resolved for the adapted grid cases.

Pressure profiles at $x = 0.7m$ are shown in Fig. 4.6. The leading-edge shock is marked by a slight compression, which is followed by a sharp compression due to the ramp shock. Each grid point is marked by a symbol on the predicted profiles to show the migration of points into the high gradient regions. Due to the increased point density in these regions for the adapted case, the corresponding pressure profiles show improvements in shock definition over the fixed grid cases, even when compared with the fixed grid solution employing twice the number of grid points. The grid-fitted case also shows improvements over the adapted case without grid-fitting in the leading-edge shock region.

In Figs. 4.7 and 4.8, wall pressure and heat transfer coefficient distributions for adapted and unadapted cases are compared with the experimental results obtained by Holden and Moselle (1969) The pressure coefficients of Fig. 4.7 were computed in the following manner

$$C_p = \frac{P_w}{\rho_\infty V_\infty^2}$$

Heat transfer coefficients are defined by

$$C_h = \frac{\mu_w}{PrRe_\infty} \frac{\sec\theta}{\frac{1}{2}(\gamma-1)M_\infty^2 + 1 - T_w} \frac{\partial T}{\partial y}$$

where a w subscript denotes values at the wall and θ is the angle between the y-axis and a line normal to the wall. The present results show reasonable agreement

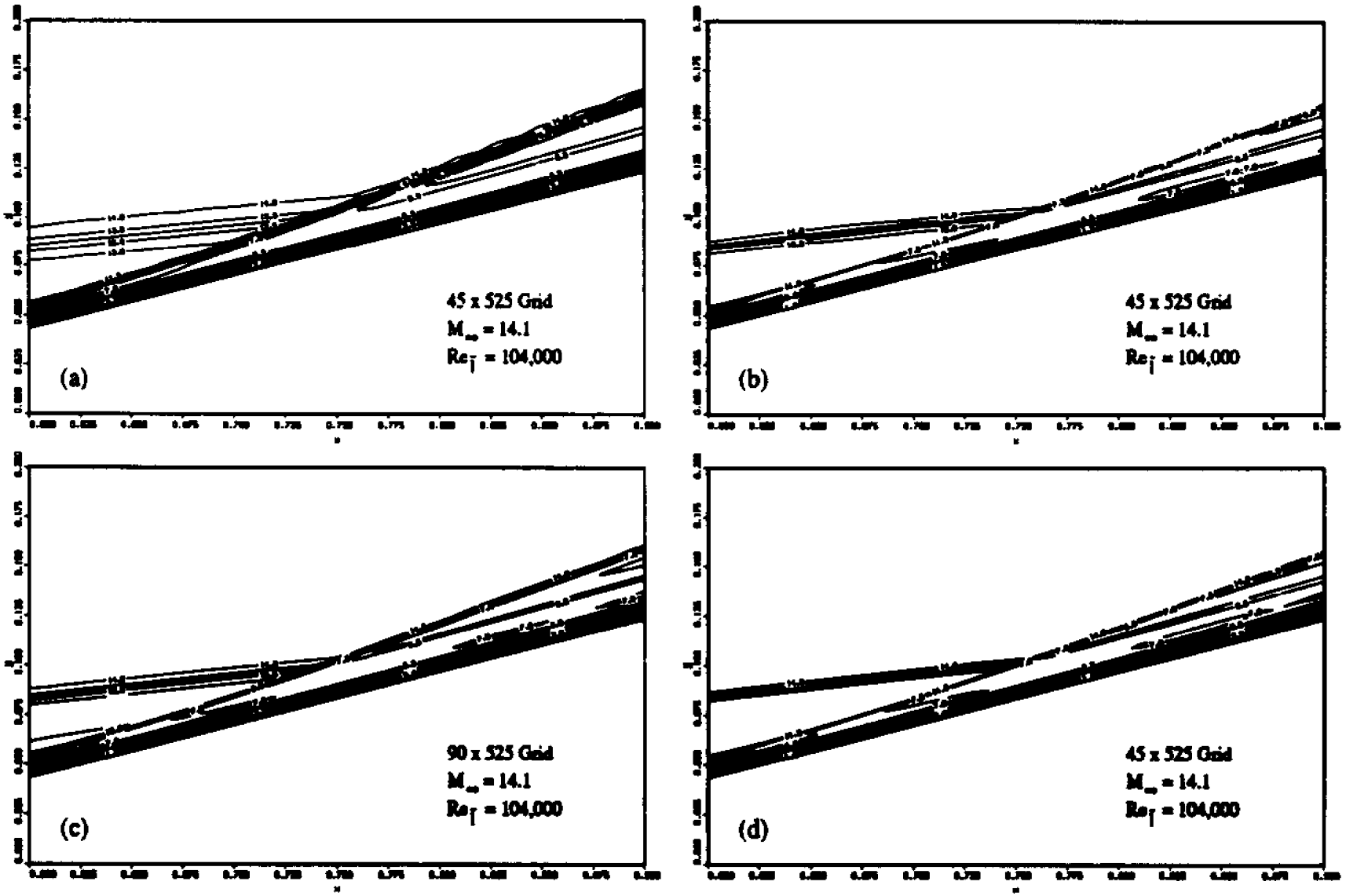


Figure 4.5: Mach number contours in the shock interaction region. (a) & (c) Without grid adaptation, (b) with grid adaptation, (d) with grid adaptation and grid-fitting.

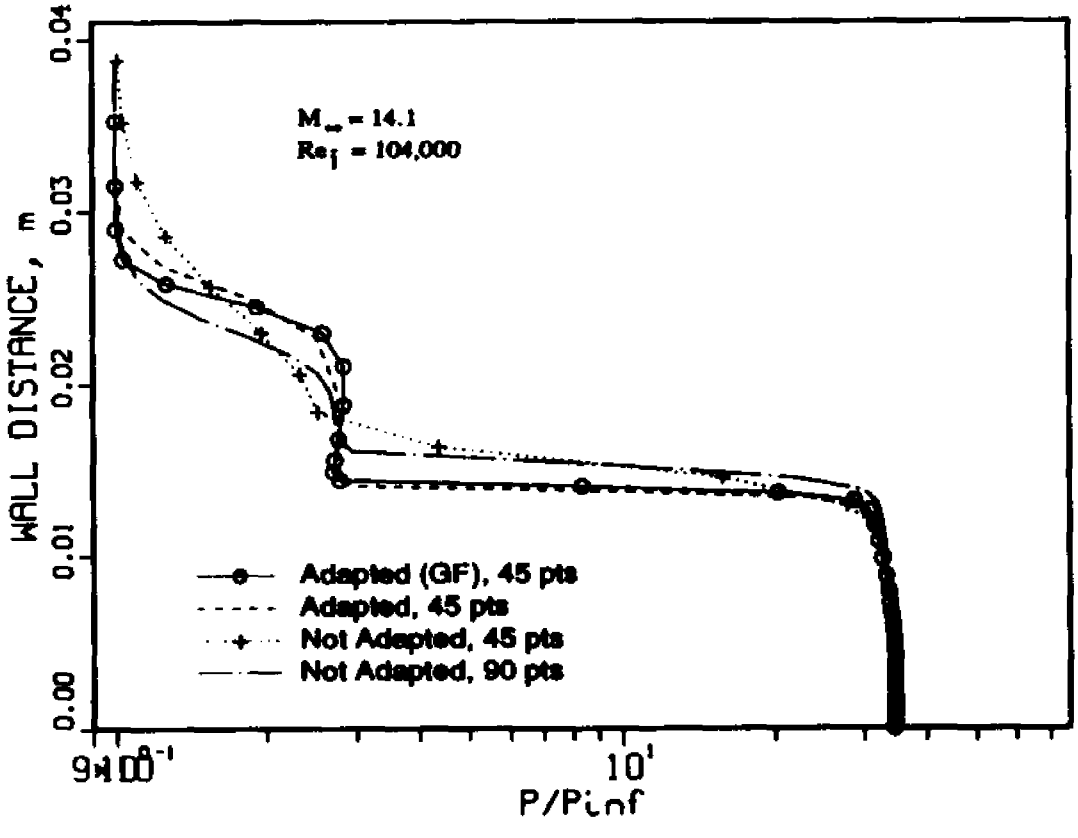


Figure 4.6: Comparison of pressure profiles (x= 0.7m).

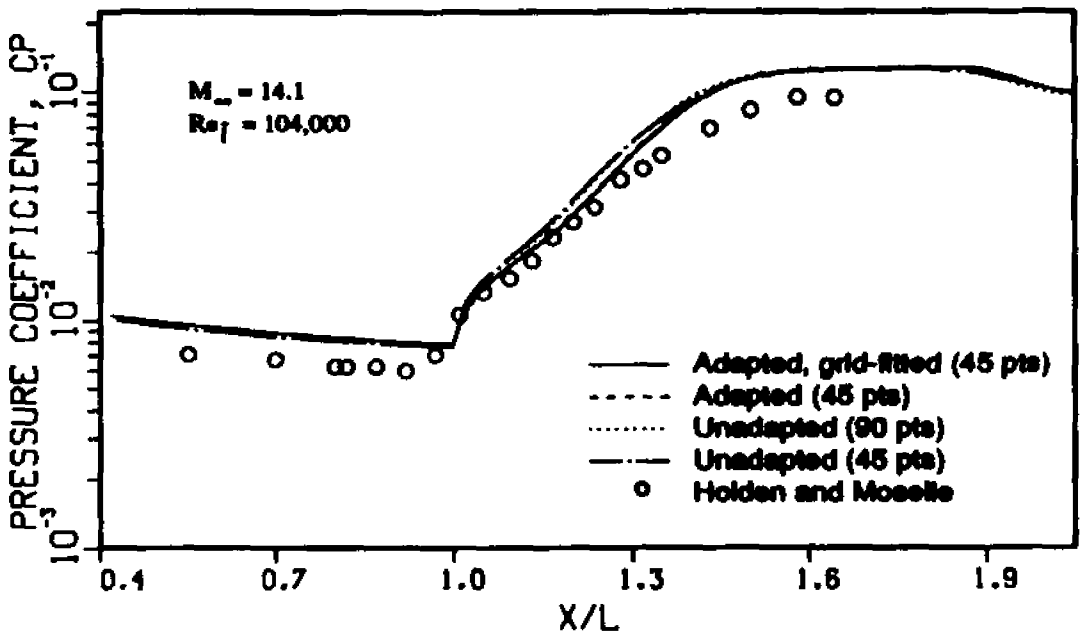


Figure 4.7: Comparison of wall pressure coefficients.

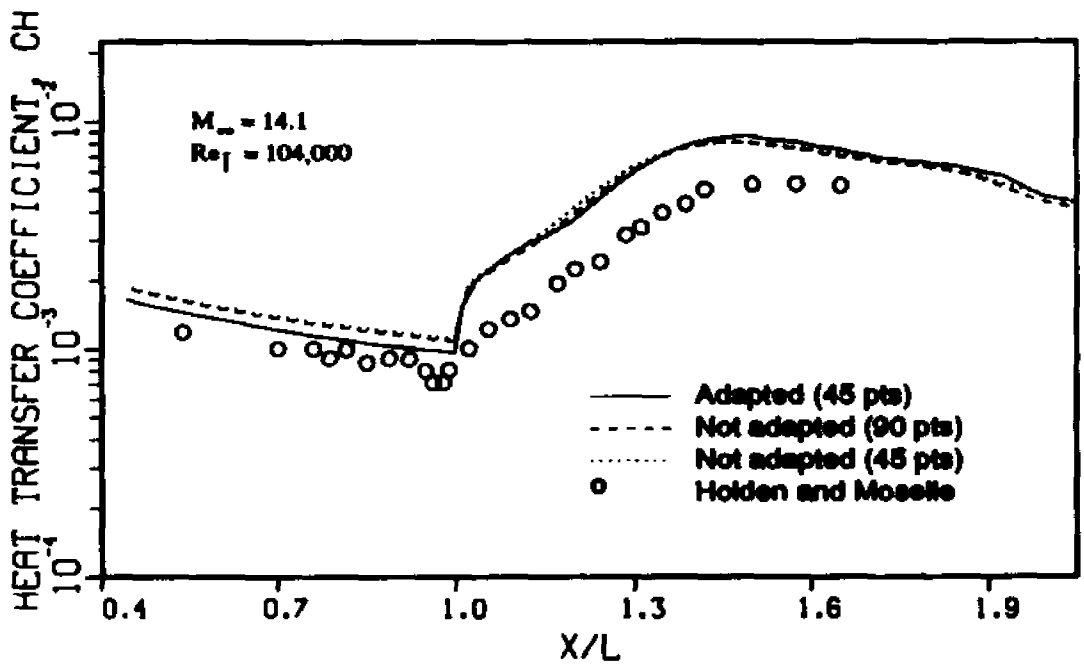


Figure 4.8: Comparison of heat transfer coefficients.

with the experimental results, although both pressure and heat transfer coefficient distributions are slightly over-estimated (Fig. 4.8). A slight improvement in pressure coefficient is noticed in the region directly above the base of the ramp. This can possibly be attributed to the slight reduction in wall spacing which occurs in the adapted cases. As expected, both pressure and heat transfer coefficient distributions for the unadapted case employing 90 grid points lie between the adapted cases and the unadapted case that employs only 45 grid points.

Computational effort was measured on a Cray Y-MP/832 computer. The unadapted case employing 45 grid points in the normal direction required 17.0 seconds of CPU time while the adapted cases required 36.4 seconds. The unadapted case employing 90 grid points used 34.9 seconds of CPU time. The use of the grid-fitting algorithm added no significant effort to the computations of the solution-adapted case. The unadapted case employing 90 grid points and the adapted case employing 45 points used approximately the same CPU time; however, shock wave resolution is significantly better for the adapted cases.

4.2 Hypersonic Inlet

A planar, cowl-type inlet geometry was studied to further examine performance of the grid adaptation procedure. This geometry (Fig. 4.9) is composed of two compression surfaces followed by a narrow inlet channel. The inviscid flowfield consists of two shock waves which impinge on the cowl surface at a common point.

The resulting solution-adapted grid is illustrated in Fig. 4.10. The cowl-surface begins at $x = 0.9m$ (Grid points in the freestream, upstream of this x-location in Fig. 4.10, have been omitted). Since this case represents an internal flow, the outer grid-line adjustment technique is not appropriate and was not employed here. This case was chosen to illustrate the ability of the present adaptation method to resolve shock wave and shock-boundary-layer interactions.

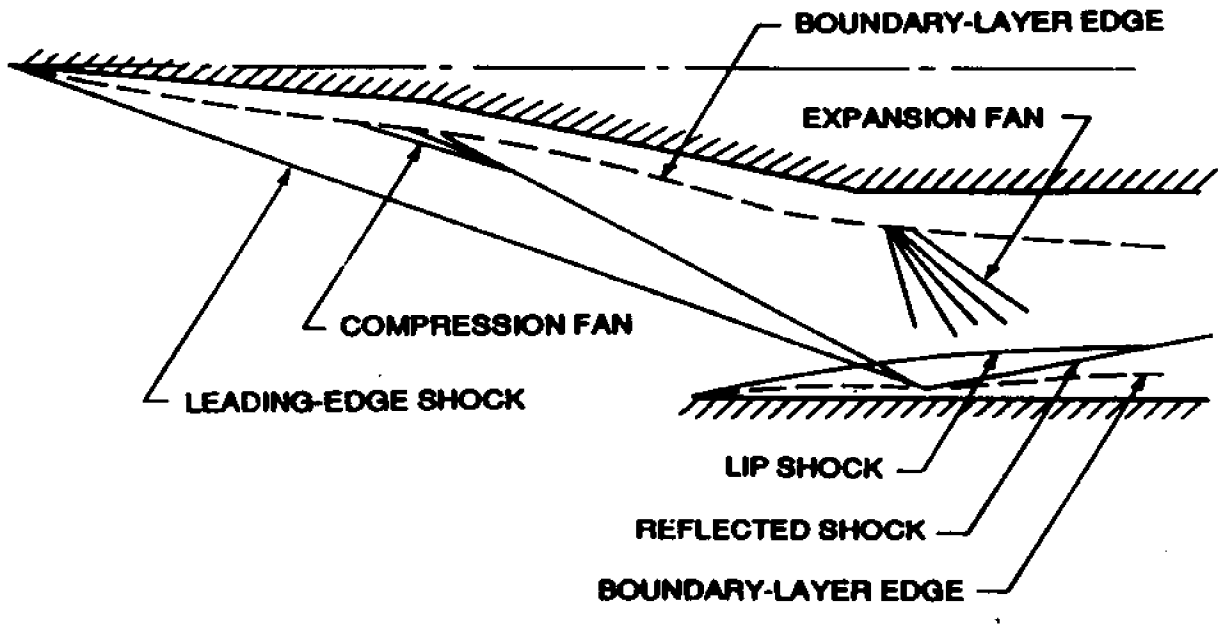


Figure 4.9: Hypersonic inlet geometry.

The freestream conditions are

$$M_\infty = 6.5 \qquad \tilde{T}_\infty = 702K \qquad \gamma = 1.4$$

$$Re_{\tilde{l}} = 2 \times 10^5 \qquad \tilde{T}_w = 1216K \qquad Pr = 0.72$$

Where $\tilde{l} = 1m$ is the distance to the downstream end of the second compression surface. No-slip boundary conditions were applied along the entire upper wall while, at the lower boundary, freestream conditions were applied on the forebody (i.e., upstream of the inlet plane located at $x = 0.9m$). At the inlet face, no-slip conditions were introduced. The grid consists of 80 grid points clustered at both upper and lower flow-field boundaries using a hyperbolic tangent stretching function [Thompson et al., 1985b]. In order to provide grid continuity at the inlet plane, grid stretching was extended the entire length of the geometry; however, to clarify the inlet plane location, only 60 grid points are shown on the forebody in Fig. 4.10.

Solution-adaptive gridding was initiated at $x = 0.04m$. On the forebody the adaptation domain consisted of the 60 points nearest the upper wall. At the inlet face location ($x = 0.9m$) no-slip boundary conditions were introduced at the cowl lip and the adaptation domain was extended to include the entire calculation domain. This procedure maintained a sufficient grid clustering at the lower boundary to resolve the viscous effects present at the cowl lip. The enlarged region of Fig. 4.10 shows the grid surrounding the shock wave interaction at $x \approx 1.0m$.

The pressure field in the region directly downstream of the inlet plane is shown in Fig. 4.11 for both unadapted and adapted cases. The two shocks can be clearly seen impinging on the boundary layer which is developing on the cowl surface. These two shock waves penetrate the lip shock, deflecting it downward slightly, and then meet at the wall where they reflect as a single, stronger shock. The resulting reflected shock travels downstream, interacting with the lip shock

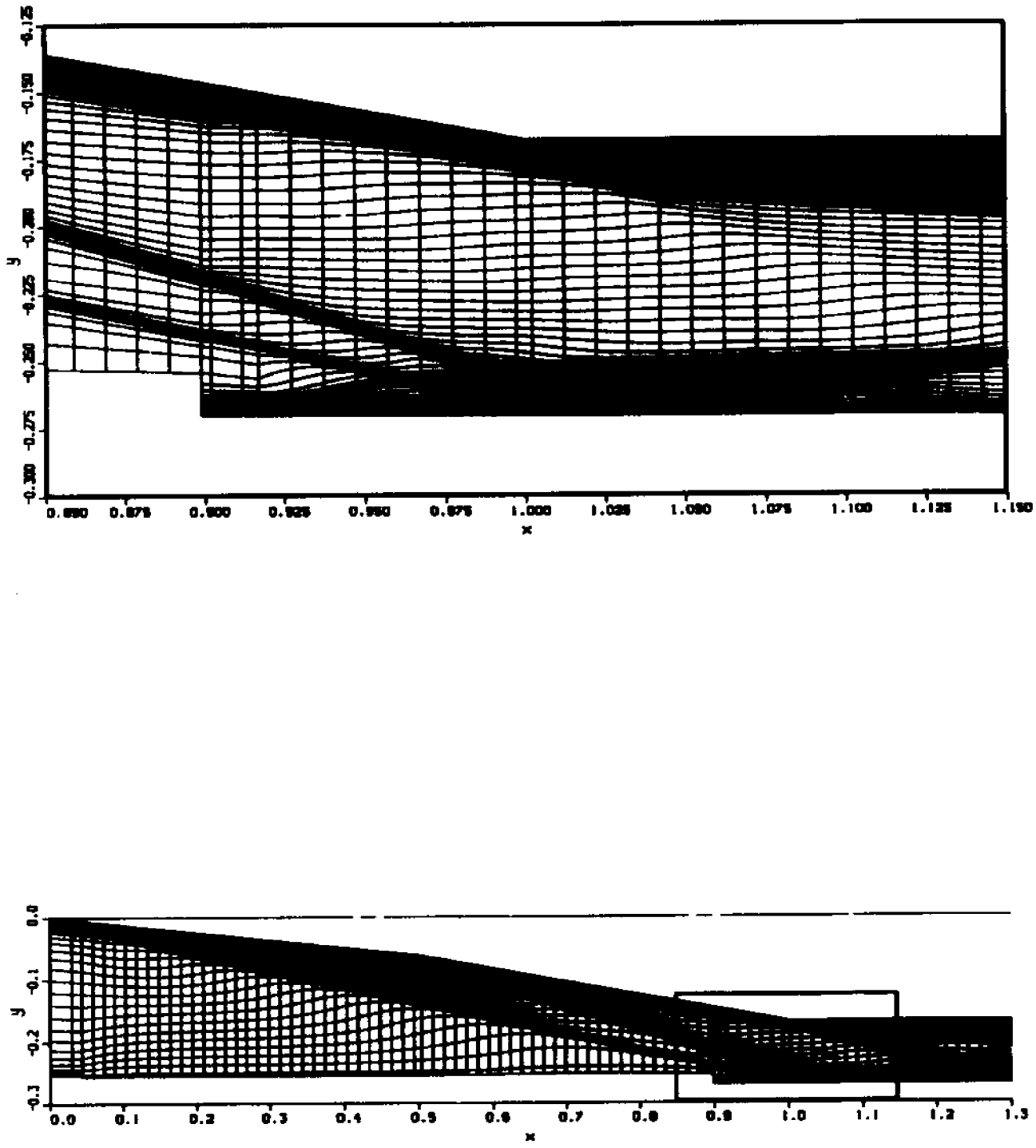
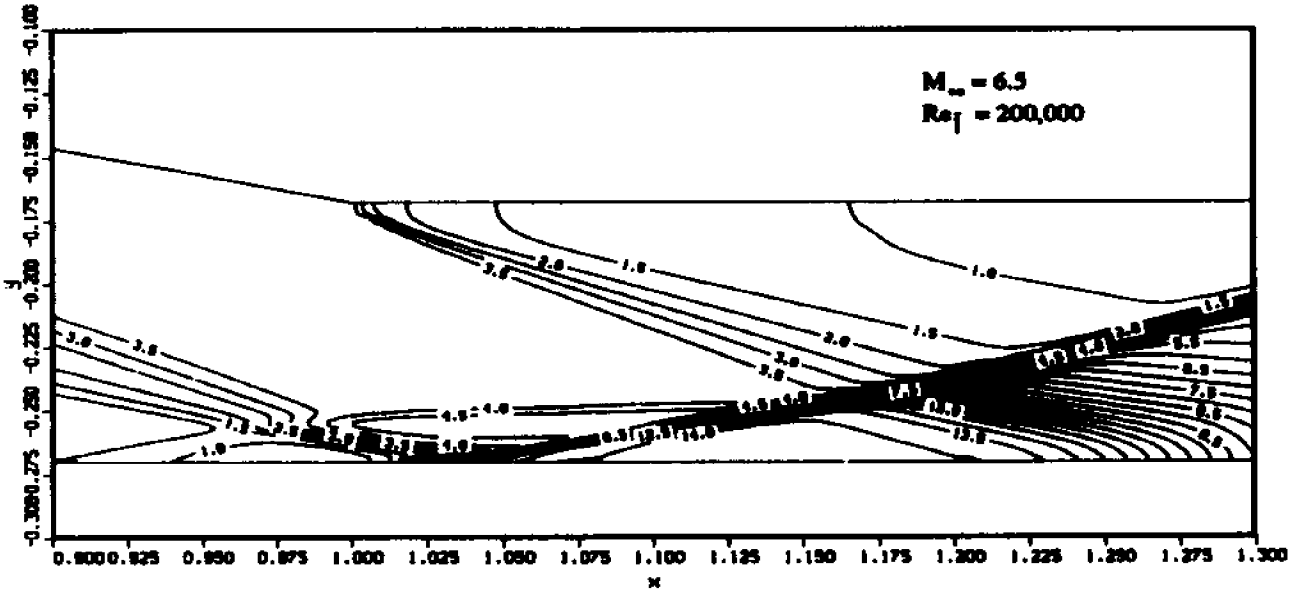
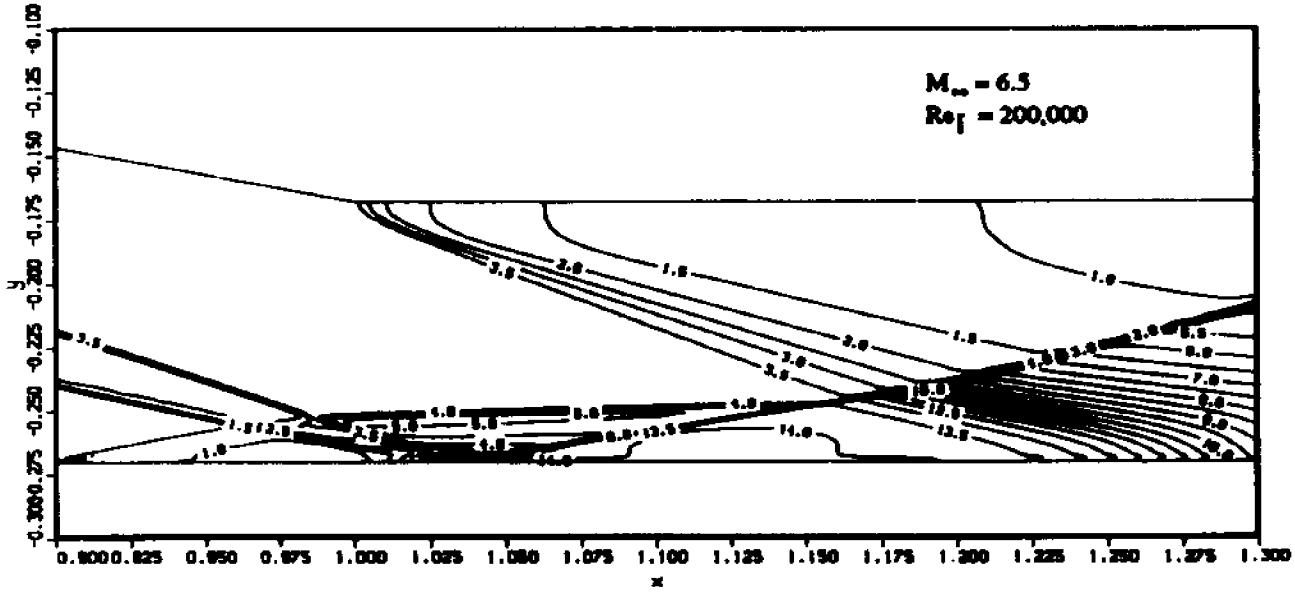


Figure 4.10: Computational grid for planar-inlet configuration.



(a)



(b)

Figure 4.11: Comparison of computed pressure contours. Planar inlet (a) unadapted and, (b) adapted.

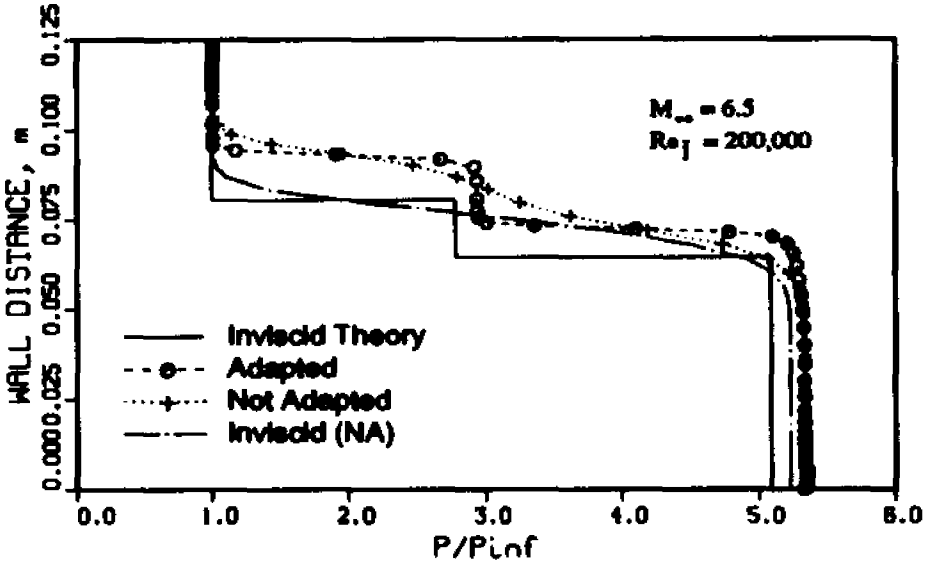


Figure 4.12: Pressure profiles (x= 0.9m).

again at $x = 1.15m$, and then is deflected by the expansion fan formed by the top of the compression surface. These characteristics are much more clearly defined in the adapted case (Fig. 4.11*b*).

Static pressure profiles at $x = 0.9m$ are shown in Fig. 4.12. Both adapted and fixed grid solutions are compared with inviscid, oblique-shock theory. The two shock waves which cross this x-location are difficult to distinguish from one another for the fixed grid case. The shape of the pressure profile for the adapted case agrees well with the shape of the inviscid-theory curve in that both shock waves are clearly defined. To verify that the upward shift in profiles for the computational results is attributed to viscous effects, an inviscid fixed-grid solution was computed. A comparison of these results clearly shows that the discrepancy between the viscous numerical solution and that of inviscid theory is due primarily to viscous effects and is not the result of numerical error.

4.3 Cone-cylinder

The final test case involves supersonic flow over a cone-cylinder arrangement. This test case is related to the study of sonic boom predictions. In sonic boom predictions it is desirable to predict the far-field pressure impulse caused by a moving shock wave. It is often difficult to obtain a satisfactory grid for this type of problem because the primary region of interest is a great distance from the body, making the solution domain very large. Conventional grid point clustering in the far-field is often not possible without over-populating other less important regions of the flow-field, resulting in wasted time and effort. Thus, the use of an adaptive grid scheme for this case is virtually imperative for obtaining an accurate solution.

For the present study, the geometry (Fig. 4.13) consists of a 3.24° half-angle cone joined by a slender cylinder. The flow is axisymmetric and calculations were performed using the PNS solver, neglecting viscosity, at a freestream Mach

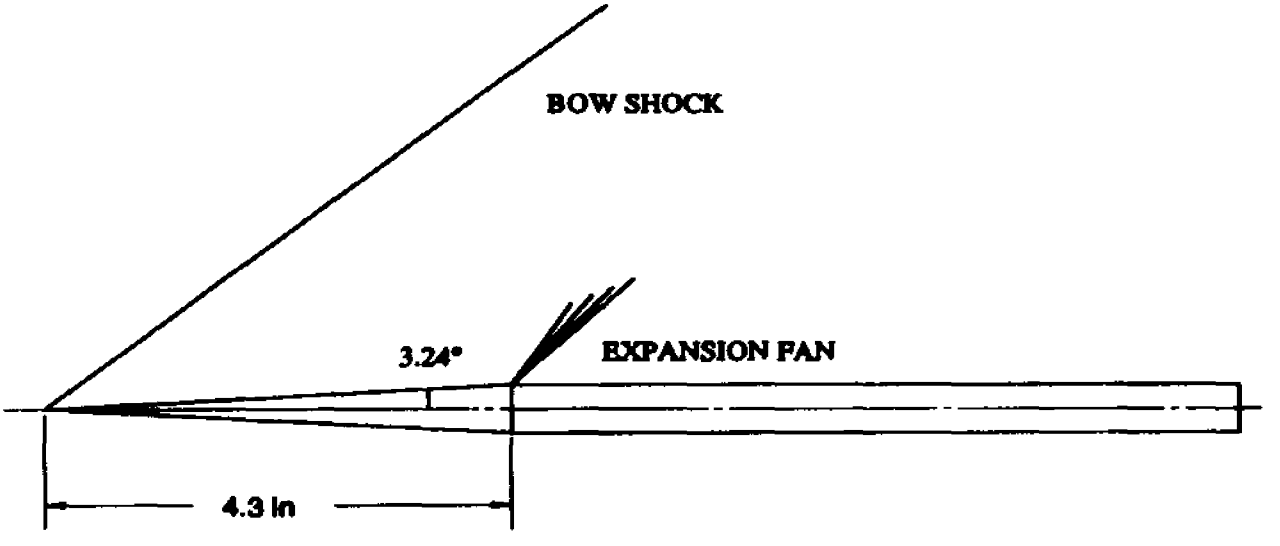


Figure 4.13: Cone-cylinder geometry.

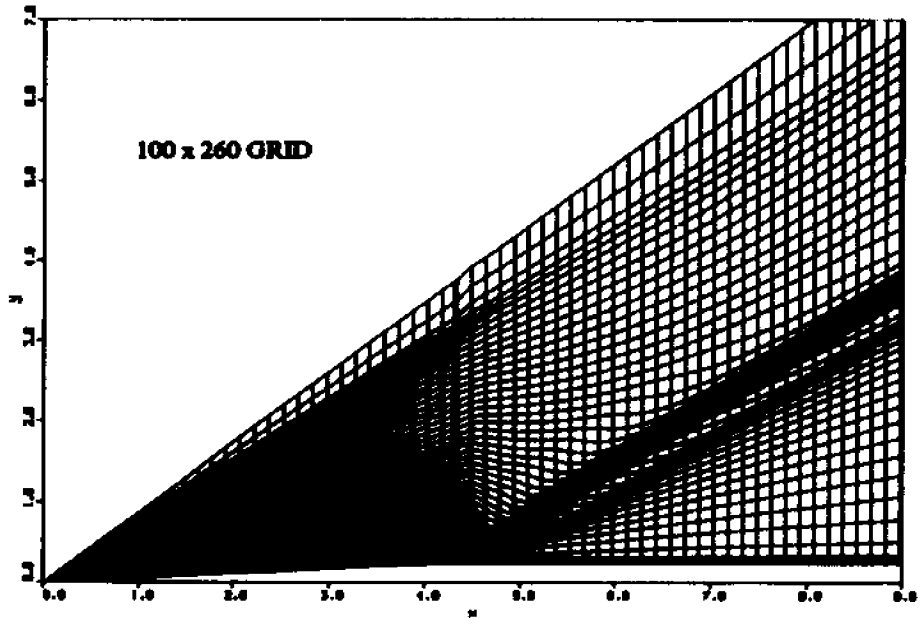


Figure 4.14: Adapted grid for cone-cylinder.

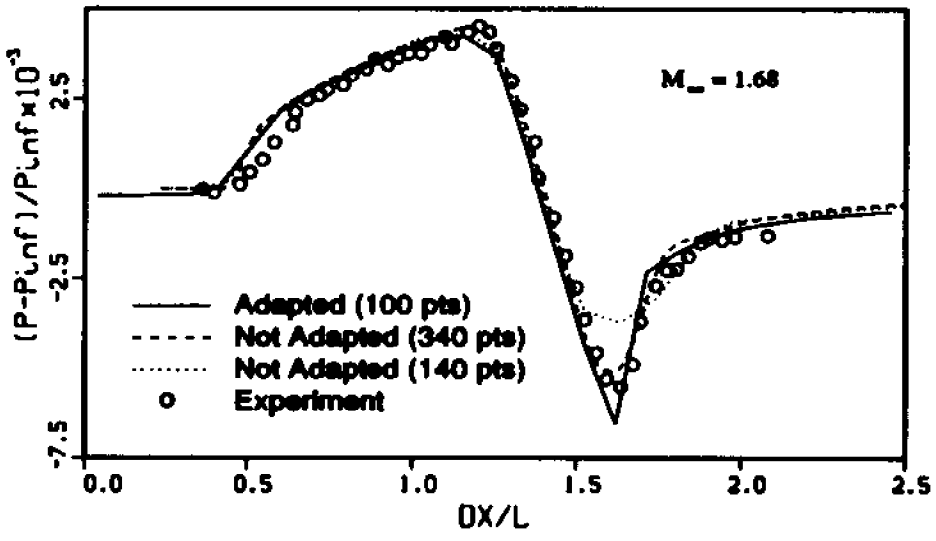


Figure 4.15: Comparison of pressure signatures.

number of 1.68. For the solution-adaptive case, 100 points were used normal to the body. A conical step-back procedure was started from freestream conditions at $x = .01in$ until a satisfactory conical starting solution was obtained. The solution proceeded from the starting plane to a distance of 130 inches downstream. This x-distance corresponds to the distance where the bow shock is at an altitude of 10 cone lengths above the body axis.

In addition to the bow shock, the flow-field features include a strong over-expansion at the base of the cone followed by a recompression wave. These features are reflected in the resulting adapted grid of Fig. 4.14. In this figure, every other grid line is shown in the radial direction. Figure 4.15 compares experimental pressure measurements at an altitude of 10 cone lengths with computed results using both adapted and fixed grids. The adapted case using 100 points is seen to be superior to even the finest unadapted case. This clearly illustrates the benefit of an adapted grid scheme for use with sonic boom calculations where the far-field flow structure is of major importance. The fixed grid case using 340 grid points required 22.5 minutes of CPU while the adapted case employing 100 grid points required 15 minutes.

4.4 Comparison of Computational Effort

A summary of computational effort for each test case is given in Table 4.1. All computations presented here where performed on a Cray Y-MP/832, and CPU times are measured in seconds.

Table 4.1: Computational effort.

CASE	GRID	#PTS	CPU
Compression corner	Fixed	90	35
	Fixed	45	17
	Adapted	45	36
Inlet	Fixed	80	170
	Adapted	80	345
Cone-cylinder	Fixed	340	1320
	Adapted	100	880

4.5 Weighting Function Comparisons

The primary focus of this subsection is to study the weighting function selection procedure presented in subsection 3.2 and compare results obtained using this technique with other common weighting function selection methods. A popular choice for a measure of the solution error is the gradient or curvature of a selected dependent flow variable or a linear combination thereof. In this subsection, comparisons will also be made on the effect of the choice of flow-field variable gradient and/or curvature used to represent the weighting functions.

For many problems in fluid flow, one dependent variable may be changing rapidly in certain regions of the flow-field, while in another region, a different dependent variable may undergo large changes. Thus, the choice of a single dependent flow variable that provides a sufficient representation of the solution error in all regions of the flow-field becomes a difficult task. A choice of pressure or density as the error estimator will adequately represent shock waves but will overlook any existing shear layers or contact surfaces.

The flow property, f, described in the last chapter, represents the solution error along the current adaptation line. The function of Eq. (3.2) is to relate the maximum and minimum grid spacings to user-requested values as well as to provide additional smoothing. Ideally, it is desired to minimize the truncation error and distribute the error uniformly over the computational domain. Although the truncation error cannot be estimated accurately, it is generally proportional to $(\Delta s)^n \partial^n \phi / \partial s^n$, where n denotes the order of accuracy.

Method I: The first method is probably the most commonly used representation of weighting functions and involves a linear combination of the gradient and the curvature of a single dependent flow variable such as pressure or density. This combination can be expressed as

$$f_i = \alpha\left(\frac{\partial \phi}{\partial s}\right) + \beta\left(\frac{\partial^2 \phi}{\partial s^2}\right) \tag{4.1}$$

where α and β are user specified, ϕ is the flow variable and s denotes differentiation along the computational coordinate line.

Method II: The next method is similar to the first in that a linear combination of gradients and curvature of a selected flow-field variable is used for representation of the weighting functions. However, additional dependent flow variables can be added in the following manner

$$f = \alpha_1\left(\frac{\partial \phi_1}{\partial s}\right) + \beta_1\left(\frac{\partial^2 \phi_1}{\partial s^2}\right) + \alpha_2\left(\frac{\partial \phi_2}{\partial s}\right) + \beta_2\left(\frac{\partial^2 \phi_2}{\partial s^2}\right) \tag{4.2}$$

Method III: As in many of the studies on solution adaptive gridding, a linear combination of gradients and curvature of selected dependent and/or physical variables is used here as weighting functions. Finding a flow-field variable which will consistently represent solution error in all regions of the solution domain is

difficult, if not impossible. As noted earlier, some variables change very rapidly in certain regions but remain fairly smooth in other regions where another variable could be undergoing severe change. In the present study, an algorithm is developed that automatically chooses which variables are to represent the weighting function at each grid point location. The scheme computes gradients and/or curvature of each user-specified variable and then, after normalizing each of these, chooses the largest in magnitude to represent the weighting function at each point. This method has proven superior to conventional weighting function selection processes, especially where large changes in one dependent variable exist in certain regions of the flow, but for another variable these large changes exist in different regions.

As presented in section 3.2, f_i is defined by Eq. (4.1) above. However, unlike method I, the gradients and curvatures are chosen from those of a number of selected variables in the following manner.

$$\frac{\partial \phi}{\partial s} = \max_k \left| \frac{\partial \phi_k}{\partial s} \right|_i, \qquad \frac{\partial^2 \phi}{\partial s^2} = \max_k \left| \frac{\partial^2 \phi_k}{\partial s^2} \right|_i$$

for all specified ϕ_k's, i.e. Mach number, pressure, density etc.

For each of the above methods studied in this section, the values of f are first normalized by

$$\bar{f}_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{4.4}$$

The weighting functions are then computed using Eq. (3.2), re-written here for convenience.

$$w_i = 1 + A\bar{f}_i^B \tag{4.5}$$

The hypersonic compression corner flow of subsection 4.1 is now used to compare the above three choices of weighting function. The flow-field conditions are identical to those presented in section 4.1. For this study, Mach number and pressure

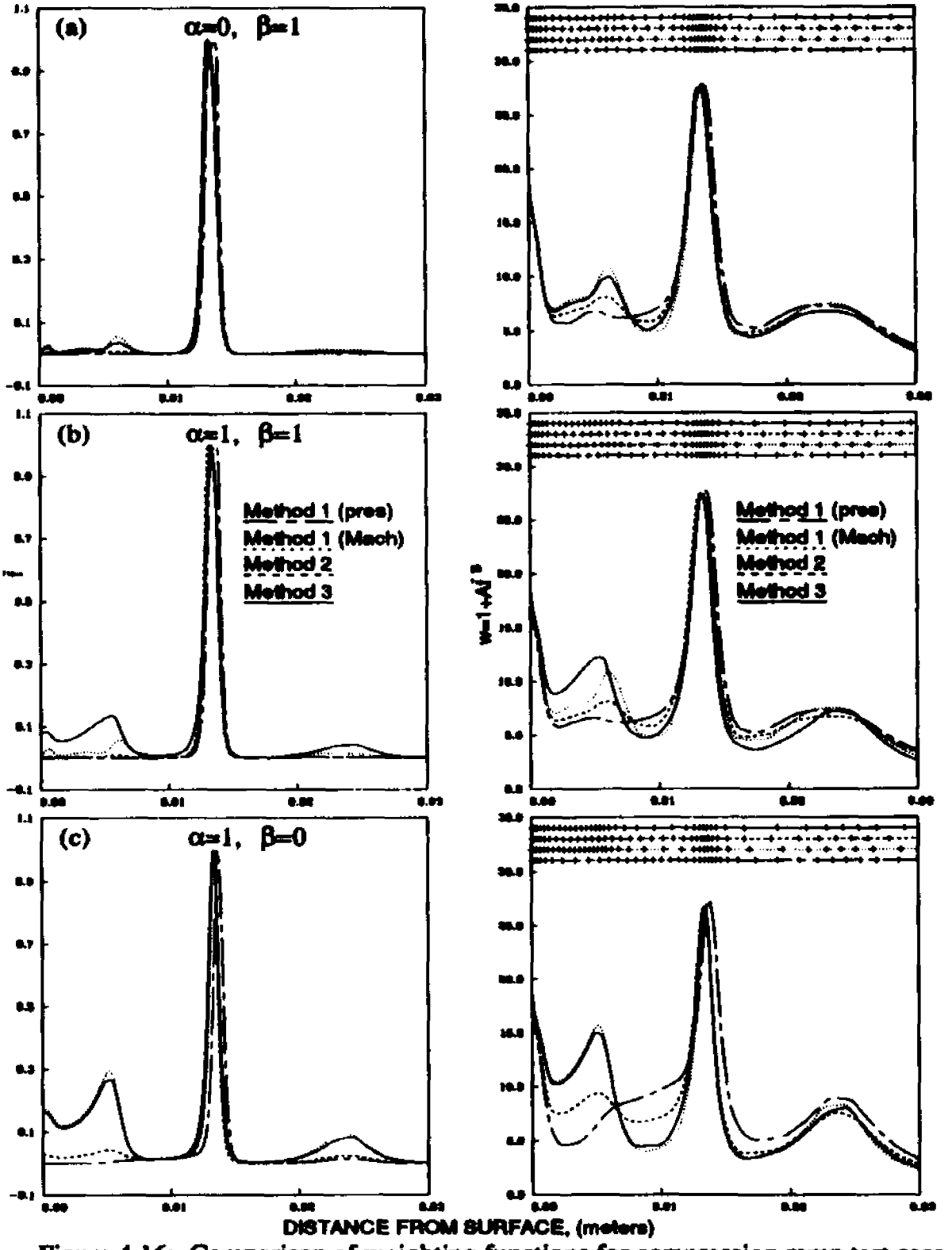


Figure 4.16: Comparison of weighting functions for compression ramp test case. (M_{∞}= 14.1, $Re_{\infty,L}$= 1.04×10^5, x= .7m, Δs_{MAX}=4.5, Δs_{MIN}=.15)

are the flow-field variables considered in methods II and III above. Method I is performed using Mach number and pressure separately. In Fig. 4.16, these four cases are compared for three different combinations of α and β. In the left side of Fig. 4.16, the quantity $\bar{f}$ of Eq. (4.4) is plotted versus distance from the surface of the ramp at an x-location of $0.7m$. The corresponding weighting function, w_i, given by Eq. (4.5) is plotted on the right side. This x-location corresponds to a location directly upstream of the shock interaction region.

At the top of each graph of of the weighting function of Fig. 4.16 the resulting grid point distributions for each case are illustrated, each tic mark denoting the position of a grid point along the resulting adapted grid line. The large spike of each graph indicates the position of the ramp shock, while the smaller peak to the right indicates the position of the leading edge shock. The peak to the left (near the wall) in the graphs of $\bar{f}$ is due to the presence of the boundary layer. Note that this feature is not as apparent in the curve of method I using pressure only. However, this feature shows up well in the curve of method I using Mach number only. The curve of method III possesses the largest changes in weighting function magnitude along the overall distribution. Near the wall, the weighting function selection procedure of method III appears to be choosing Mach number to represent $\bar{f}$, while the procedure of method II uses an average of both Mach number and pressure.

The spike at the wall in the graphs of the weighting function (left side), is not strongly evident in the graphs of $\bar{f}$. This spike in the weighting function, which is of the same magnitude for all 12 cases, is due to modification procedure by the adaptive grid algorithm near the wall in order to meet the user-requested wall spacing. This process was described in Chapter 3. Note how the weighting function as defined in Eq. (4.5) has an amplifying effect on the solution error distribution.

In Figs. 4.16*a* the magnitudes of $\bar{f}$ and w are computed using curvature only. Figs. 4.16*c* show the resulting distributions of $\bar{f}$ and w using only flow-field gradient, while Figs. 4.16*b* show distributions resulting from equal consideration of flow-field gradient and curvature. As expected, the use of curvature produces a smoothing effect on overall solution error distributions. Peaks are wider when defined using curvature, resulting in slightly thicker grid point clustering regions (note the top of the graphs of the weighting function).

In comparing the grid point distributions, shown at the top of the figures, it is easily seen that the ramp shock is always well resolved. The near wall viscous region is also well resolved in view of the user-specified near wall spacing requirement. However, differences in the resolution near the boundary layer edge are evident in the different weighting function selection techniques. Method I, with Mach number as the variable used for adaptation, and Method III, which is the practice adopted in this work, show the best resolution in the boundary layer region. Method I has the disadvantage that the user cannot specify the variable of adaptation, and in many applications it is difficult to determine, *a priori*, which variable is the important one in a given region. Method III removes this drawback and the need for an *a priori* selection of the adaptation variable of interest. Due to these advantages of method III, it has been adopted in the present work.

CHAPTER 5

Three-Dimensional Adaptive Grid Procedure

This section presents the extension of the two-dimensional line-by-line solution-adaptive technique described in the previous section to a plane-by-plane grid adaptation procedure for three-dimensional spatially parabolic flows. The three-dimensional solution-adaptive marching procedure begins by first obtaining a preliminary solution at the current marching station. Based on this preliminary solution, high-error estimate regions are first identified. Next, the grid surface at this marching station is adapted in both cross-flow directions, re-clustering grid points into the high-error regions. The PNS solver then re-steps over the refined grid at this marching station to obtain an improved solution for the flow-field. The adapted grid is then projected downstream to the next marching station, and the procedure is repeated. This chapter outlines the details associated with this basic procedure.

5.1 Three-Dimensional Methodology

As described earlier for the two-dimensional method of Chapter 3, the three-dimensional grid point re-distribution process is broken into a sequence of uni-directional adaptations along each fixed coordinate line. However, for extension to a three-dimensional spatially parabolic flow, grid adaptation must occur in both cross-flow directions. This is done by re-distributing grid points along each coordinate line in each of the two cross-flow directions. If this adaptation process is to be successful, additional information must be included in the grid adaptation equations in order to maintain a sufficient degree of grid orthogonality and smoothness in the additional cross-flow direction.

For each grid point along the current adaptation line, the three-dimensional grid adaptation process includes the modification of Eq. (3.4) to include a torsional dependence from two different directions. One torsion spring extends from the adjacent coordinate line directly upstream of the current adaptation line in the same manner as for the two-dimensional technique, and the other torsional influence is projected from the neighboring grid point in the cross-flow direction (See Fig. 5.1). To distinguish between the two, an 's' subscript denotes streamwise torsional quantities while a 'c' subscript will denote quantities involving cross-flow torsional dependence.

Using the above described notation and performing a force balance along a constant ζ-line on the resulting system of tensional and torsional springs about the point i, j in Fig. 5.1, the following system of equations is obtained,

$$w_i(s_{i+1} - s_i) - w_{i-1}(s_i - s_{i-1}) - K_{s_i}\theta_{s_i} - K_{c_i}\theta_{c_i} = 0 \qquad (5.1)$$

where s_i is the length of the current adaptation line up to the point i, j in Fig. 5.1, and K_{s_i} and K_{c_i} are proportionality constants. The torsion angles, θ_{s_i} and θ_{c_i}, can be approximated using the notation of Fig. 5.1 as $\theta_{s_i} \approx (s_i - s'_{s_i})/|EC|$

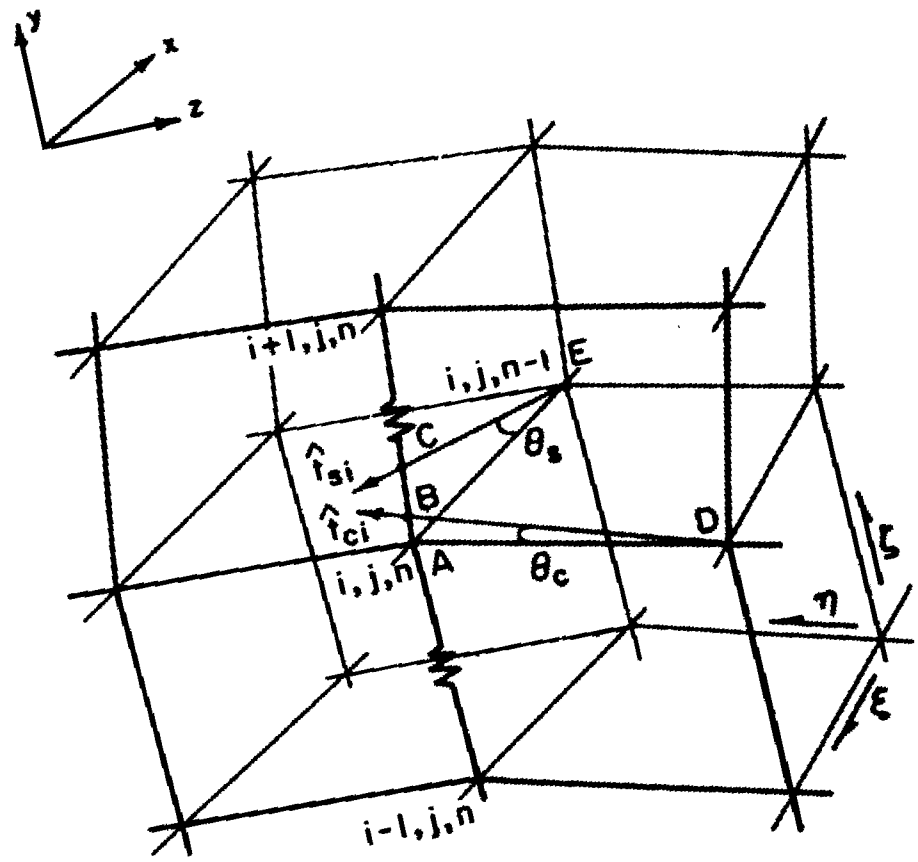


Figure 5.1: Three-dimensional adaptive mesh geometry.

and $\theta_{ci} \approx (s_i - s'_{ci})/|DB|$. The quantity s'_{si} is defined as the total length of the coordinate line up to the location of its intersection with the streamwise torsion vector, $\hat{\mathbf{t}}_{si}$. This point corresponds to point C in Fig. 5.1. The quantity s'_{ci} is defined similarly and is the corresponding intersection point of the cross-flow torsion vector, $\hat{\mathbf{t}}_{ci}$, with the current adaptation line and is denoted in Fig. 5.1 by the point B. Substitution of these expressions for θ_{si} and θ_{ci} into Eq. (5.1), rearranging and defining $\tau_{si} = K_{si}/|EC|$ and $\tau_{ci} = K_{ci}/|DB|$, the following linear system is obtained:

$$w_{i-1}s_{i-1} - (w_{i-1} + w_i + \tau_{si} + \tau_{ci})s_i + w_i s_{i+1} = -(\tau_{si}s'_{si} + \tau_{ci}s'_{ci}) \qquad (5.2)$$

5.2 Calculation of Torsional Parameters

Equation (5.2) indicates that as the terms τ_{si} and τ_{ci} become large, the equilibrium position of the ith point approaches the position corresponding to a weighted average of s'_{si} and s'_{ci}. These two quantities represent the equilibrium positions of the individual torsional springs projecting from neighboring grid lines, denoted by the vectors $\hat{\mathbf{t}}_{si}$ and $\hat{\mathbf{t}}_{ci}$, respectively, in Fig. 5.1. Thus, if $\tau_{si} = \tau_{ci}$ and both are much larger than the remaining terms of Eq. (5.2), the resulting solution to Eq. (5.2) becomes

$$s_i = \frac{1}{2}(s'_{si} + s'_{ci})$$

Consequently, values of s'_{si} and s'_{ci} much different from $\bar{s}_i$, the initial grid point positions, can result in excessive movement of grid points, even without any input from the error measure. In the present work, to prevent excessive skewing, large changes in the grid point positions are not desired since a new solution is obtained on the updated grid surface immediately after grid adaptation. Initially, before adaptation and during the preliminary solution process, the downstream side of the computational slab at the nth or current marching station is simply a

proportioned duplicate of the upstream side. After adaptation, these two sides of the computational slab should not differ too severely from each other to permit an improved solution during the re-marching step of the solution-adaptive procedure. If excessive skewing occurs the solution process becomes unstable. What is desired is that for even large values of τ_{si} and τ_{ci}, the grid continually retains the smoothness necessary for the solution on the adapted grid. Previous work using the current grid adaptation method have not reported the stability and accuracy link between the grid skewness and the solution on the adapted grid. In order to achieve the precise and bounded movement required in the present solution-adaptive scheme, the values of s'_{si} and s'_{ci} must remain close to the preliminary grid point positions, $\bar{s}_i$, and, at the same time, the values of τ_{si} and τ_{ci} of Eq. (5.2) must be chosen so as to hold the grid close to these locations. Only a small range of the torsion terms will accomplish the latter part of this task. Values of τ_{si} and τ_{ci} that are too large will smother the effect of the weighting functions, causing the grid to move independent of the solution, while values too small permit excessive movement. Davies et al. (1989) assumed the constants K_{si} and K_{ci} of Eq. (5.1) to be a function of the maximum weighting function and the grid cell face aspect ratio. This procedure was adopted in Chapter 3. However, in some cases, the cell face aspect ratio can vary by several orders of magnitude and, as a result of this, can produce unpredictable variations in the torsion terms. Djomehri and Deiwert (1988) assumed that the torsion terms are proportional to an average value of the weighting functions along the adaptation line. Using this assumption the resulting values of the torsion terms were found to vary inconsistently from the remaining terms of Eq. (5.1) at each point.

In the present work, a somewhat different approach is adopted. First, to make all the terms of Eq. (5.2) equal in magnitude at each point along the adaptation line, a stiffness parameter, Ω_i, is defined in the following manner:

$$\Omega_i = \lambda(w_{i-1} + w_i)\bar{\alpha}_i$$

where λ is a user-defined stiffness 'safety-factor' and is generally given numerical values close to unity, and $\bar{\alpha}_i = \alpha_i/\alpha_{min}$ is the aspect ratio of the grid cell normalized with respect to the minimum cell aspect ratio. Each aspect ratio is computed using the notation of Fig. 5.1 as follows:

$$\alpha_i = \frac{s_{i+1,j,n} - s_{i-1,j,n}}{|DB|}$$

The value of Ω_i is then divided into streamwise and cross-flow components according to the lengths of the streamwise and cross-flow torsion springs, $|EC|$ and $|DB|$, respectively.

$$K_{s_i} = \frac{|DB|}{|EC| + |DB|}\Omega_i \qquad K_{c_i} = \frac{|EC|}{|EC| + |DB|}\Omega_i$$

Substituting these expressions for the proportionality constants, K_{s_i} and K_{c_i}, into the definitions for τ_{s_i} and τ_{c_i} given above, the following expressions are obtained:

$$\tau_{s_i} = \frac{|DB|}{|EC| + |DB|}\frac{\Omega_i}{|EC|} \qquad \tau_{c_i} = \frac{|EC|}{|EC| + |DB|}\frac{\Omega_i}{|DB|} \tag{5.3}$$

The use of Eqs. (5.3) in (5.2) ensures that all terms of Eq. (5.2) are of the same magnitude for all points along the line. When the length of the streamwise torsion spring, $|EC|$, is equal to the length of the cross-flow torsion spring, $|DB|$, then the torsional influence is distributed in the two directions equally and, as a result, equal consideration is given to both equilibrium positions, s'_{s_i} and s'_{c_i}. When the two springs are of unequal length (i.e. one grid line is closer to the current adaptation line than the other), more torsional dependence develops from the closer of the two neighboring grid lines. Another advantage of defining the torsion terms in this manner is that a single parameter, λ, is involved. Other formulations of the torsion terms investigated by the present author required the

introduction of torsion parameters in each of the two directions. As a result, two parameters were required, both of which exhibited a strong dependence on the relative grid scales.

The two terms on the right side of Eq. (5.2) represent torsional forces acting on the ith grid point from the streamwise and cross-flow directions, respectively. In the absence of these terms, Eq. (5.2) is simply a statement of solution error equi-distribution along a coordinate line, and the resulting grid point distribution, s_i, satisfies Eq. (3.1) ($w_i \Delta s_i$ = constant). The addition of the torsional force terms modifies this equi-distribution concept but provides a means of relating adjacent grid lines in order to maintain a user-requested degree of grid orthogonality and straightness. Quite often the direction of these two forces can dictate opposing grid point motion. For complex problems, this behavior has been observed to introduce undesirable irregularities in the adapted grid. However, for relatively simple geometries the above formulation was found to produce satisfactory results. In later chapters, more complex geometries will be considered, and this problem will be addressed further.

A problem in the above formulation for the torsional influence arises when adapting the first grid-line of each cross-flow plane. This grid-line does not have a cross-flow neighbor. One method of resolving this problem is to temporarily set the torsional dependence in the cross-flow direction to zero when adapting the first line at each marching station. This procedure was found to introduce grid irregularities along the boundaries. Another method which is used in the present work and provides increased grid continuity throughout the grid is to introduce a fictitious grid line outside the computational domain and adjacent to the boundary line of interest. This fictitious grid line is assumed to be identical to the first interior grid line and provides a means of cross-flow support to the grid points when adapting the first line.

5.3 Determination of Reference Locations

A method of selecting s'_{s_i} and s'_{c_i} is now presented which prevents these quantities from varying significantly from $\bar{s}_i$, the initial grid point positions, so that grid skewness is prevented using the values of τ_{s_i} and τ_{c_i} formulated above. These quantities are selected in a manner so as to preserve the information that s'_{s_i} and s'_{c_i} should contain concerning the direction in which the grid should move. If the ith torsion vectors, $\hat{\mathbf{t}}_{s_i}$ and $\hat{\mathbf{t}}_{c_i}$, are allowed to cross the current adaptation line at positions greater than the point $i+1, j, n$ or less than $i-1, j, n$ of Fig. 5.1, then, for sufficiently high values of τ_{s_i} and τ_{c_i}, the grid can become distorted to the extent that the second solution step becomes unstable. The quantities s'_{s_i} and s'_{c_i} should represent the positions the ith grid point must take in order that the grid acquire the desired amounts of straightness and orthogonality in the streamwise and cross-flow directions. To provide a more useful solution-adaptive scheme, grid point motion must be further limited, allowing only incremental movement at each marching station and not permitting straightness and orthogonality constraints to be satisfied too rapidly.

Figure 5.2 illustrates the construction of the torsion vector, $\hat{\mathbf{t}}$, at two arbitrary grid locations. Since the construction for both streamwise and cross-flow contributions is similar, Fig. 5.2 may be assumed to represent either. The vector $\hat{\mathbf{n}}$ represents local grid orthogonality while $\hat{\mathbf{s}}$ is a measure of local grid straightness. The cross-flow torsion vector, $\hat{\mathbf{t}}_{c_i}$, is then defined as

$$\hat{\mathbf{t}}_{c_i} = C_c \hat{\mathbf{s}}_{c_i} + (1 - C_c)\hat{\mathbf{n}}_{c_i} \tag{5.4}$$

In Eq. (5.4) the vector $\hat{\mathbf{n}}_{c_i}$ represents local grid orthogonality and is computed as an average of the normal to the current grid line and the normal to the adjacent grid line in the cross-flow direction. The vector $\hat{\mathbf{s}}_{c_i}$ is a measure of local grid straightness and is computed as the vector from the point $i, j-2, n$ to the point

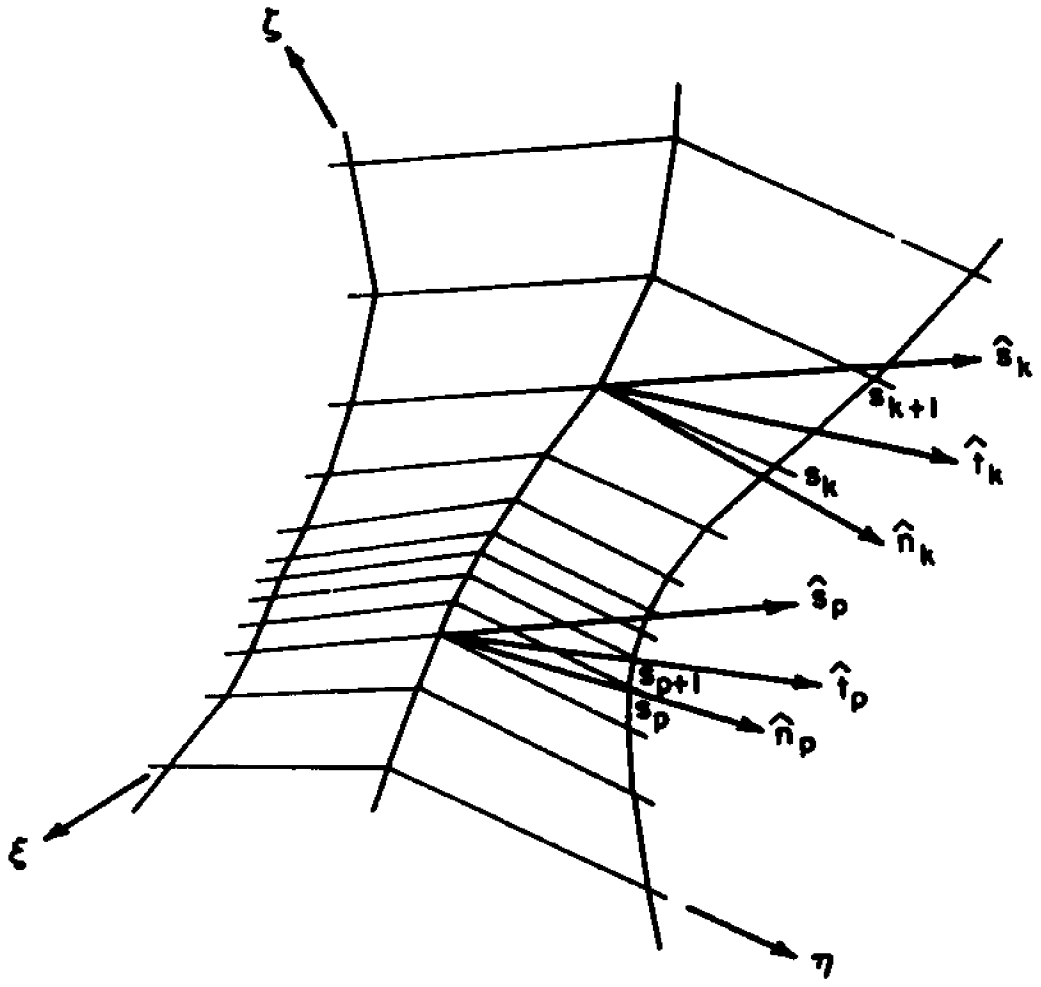


Figure 5.2: Torsion vector geometry.

$i, j - 1, n$ (point D of Fig. 5.3). The constant C_c can range from zero to unity and represents the proportion of grid orthogonality to straightness restrictions imposed on the equilibrium position of the torsional springs in the cross-flow direction. A value close to zero enforces more orthogonality while values closer to unity provide more grid straightness. As before, the c-subscript of Eq. (5.4) denotes cross-flow quantities. The straightness vector in the streamwise direction, $\hat{\mathbf{s}}_{s_i}$, is computed as the vector from the point $i, j, n-2$ to the point $i, j, n-1$ of Fig. 5.3 and is used in a similar formulation for $\hat{\mathbf{t}}_{s_i}$, the streamwise torsion vector.

The kth torsion vector in the upper portion of Fig. 5.2 crosses the current adaptation line inside the kth grid cell (i.e. between $\bar{s}_k$ and $\bar{s}_{k+1}$). Thus for a sufficiently large value of the torsion terms of Eq. (5.2) the point $\bar{s}_k$ will move, at most, to the point where $\hat{\mathbf{t}}_k$ crosses the the current adaptation line. The pth torsion vector shown in the lower part of this figure crosses the current adaptation line outside the pth grid cell and, if the surrounding torsion vectors are not properly positioned, a situation arises whereby grid lines can either become severely skewed or can even cross one-another. To remedy this, each torsion vector, $\hat{\mathbf{t}}_i$ is limited so that its intersection point is within $\frac{1}{2}\Delta s_i$ of $\bar{s}_i$ (or $\frac{1}{2}\Delta s_{i-1}$, depending on which side of the ith point $\hat{\mathbf{t}}_i$ crosses). This imposes further limitations on grid point motion, and a greater range of values for τ_{s_i} and τ_{c_i} exists for which the re-marching step of the solution-adaptive scheme remains stable.

There is no guarantee that $\hat{\mathbf{t}}_{s_i}$ will intersect the current adaptation line at all and, in general, rarely does. To illustrate this point the grid topology in Fig. 5.1 has been rotated by 90,° and the resulting new view point is displayed in Fig. 5.3. To assign an approximate value to s'_{s_i}, the vector $\hat{\mathbf{t}}_{s_i}$ is projected onto the current adaptation line in a manner which will preserve the information concerning grid characteristics supplied by the values of s'_{s_i} and s'_{c_i}. The shaded

plane of Fig. 5.3 is the plane containing the torsion vector, $\hat{\mathbf{t}}_{s_i}$, and a vector denoted as $\hat{\mathbf{p}}$ which is taken as the normal to $\hat{\mathbf{t}}_{s_i}$ and $\hat{\mathbf{r}}_k$, a unit vector in the direction of the current line segment. The index k denotes either i or $i-1$, depending on whether $\hat{\mathbf{t}}_{s_i}$ subtends above or below, respectively of the ith point along the current adaptation line (point A of Fig. 5.3). The point C is then found to be the point where $\hat{\mathbf{r}}_k$ intersects the plane of $\hat{\mathbf{t}}_{s_i}$ and $\hat{\mathbf{p}}$. As mentioned above, the distance $|AC|$ is limited to a maximum value of $\frac{1}{2}\Delta s_k$, and if it exceeds this value, it is reset to $\frac{1}{2}\Delta s_k$.

For the present work, a constant ξ-surface corresponds to a constant x-plane in physical space, which is planar and the line segments $|DB|$ and $|AB|$ are easily found from the simultaneous solution of the y- and z-components of the following vector equation:

$$|DB|\hat{\mathbf{t}}_{c_i} = \hat{\mathbf{DA}} + |AB|\hat{\mathbf{r}}_k$$

where $\hat{\mathbf{DA}}$ is a vector from point D to point A of Fig. 5.3, $\hat{\mathbf{t}}_{c_i}$ and $\hat{\mathbf{r}}_k$ are unit vectors in the indicated directions and, as before, the index k denotes i or $i-1$, depending on whether $\hat{\mathbf{t}}_{c_i}$ crosses above or below the ith point. In general, the planar assumption is not required, and if not present, a three-dimensional analysis similar to that described above for evaluating $\hat{\mathbf{t}}_{s_i}$ would be needed to evaluate the intersection point of $\hat{\mathbf{t}}_{c_i}$ and $\hat{\mathbf{r}}_k$. To further limit point movement and increase the stability of the improved solution step for a wider range of τ_{c_i}, the distance $|AB|$ of Fig. 5.3 is also limited to $\frac{1}{2}\Delta s_k$.

Once the points B and C of Fig. 5.3 are located, s'_{s_i} and s'_{c_i} are computed as the arclength of the adaptation line up to these points. These values of s'_{s_i} and s'_{c_i} provide Eq. (5.2) with an additional force acting on the grid points, pulling them in the direction they must move in order that grid straightness and orthogonality requirements be satisfied. Equation (5.2) is tri-diagonal and can then be easily solved for s_i, the final grid point positions. This is done in an

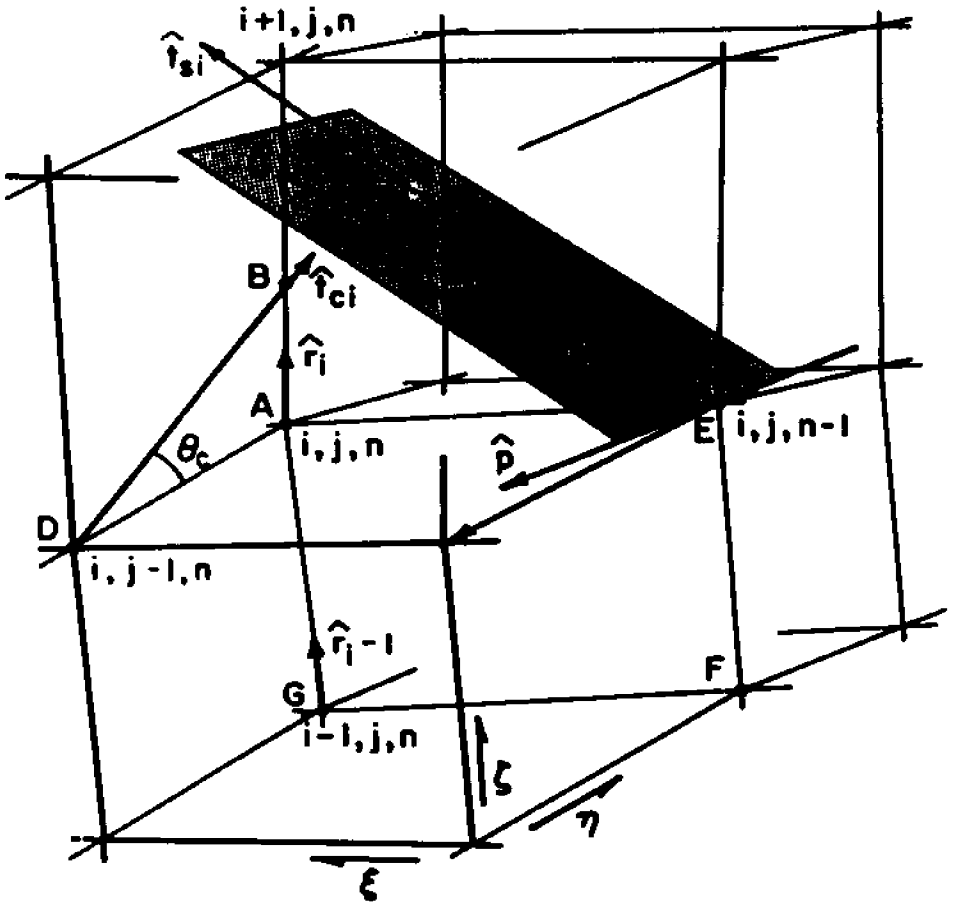


Figure 5.3: Determination of torsion vector locations.

iterative manner, interpolating the weighting functions at each iteration until the grid becomes stationary.

5.4 Adaptation Boundaries

The adaptation domain can include all or be a subset of the entire solution domain. Special procedures are necessary at the boundaries of the adaptation domain so as to allow a smooth transition of the grid into the adaptation domain. At each adaptation line a specified edge spacing, Δs_{edge} is defined as either a required wall spacing if the adaptation domain is adjacent to a solid boundary or as the grid spacing immediately outside the adaptation domain if it is a subset of the entire domain. Once Δs_{edge} is specified, a few simple modifications to Eq. (5.2) are imposed to ensure that the resulting edge spacing meets this requirement. First, s'_{s1} and s'_{c1}, the equilibrium positions of the torsion springs at the boundary are reset to Δs_{edge}. The stiffness factor, λ, is increased slightly in this region to hold the points at these positions. To provide a smooth transition at the boundary the differences, $s'_{s1} - \Delta s_{edge}$ and $s'_{c1} - \Delta s_{edge}$, are then distributed into the adaptation domain by adjusting s'_{si} and s'_{ci}, respectively, in a manner so as to ensure that they remain monotonically increasing. A problem with this edge treatment of the grid reference position is that it does not preserve grid smoothness. A method of adjusting the surrounding reference positions so that the resulting distribution is monotonic and smooth is very difficult.

Another method of resolving spacing discontinuities at the boundaries of the adaptation domain, which has been found to provide improved results over those obtained by tapering with the grid reference positions, is to adjust the weighting functions in this region. Since the product $w_i \Delta s_i$ is assumed constant along the grid line, the edge weighting function, w_1, is redefined as $w_1 = P/\Delta s_{edge}$, where P is an average product $w_i \Delta s_i$ over the grid line. The surrounding weighting functions are then adjusted to provide a smooth transition. To make this change

effect the grid spacing, the torsional parameters, λ, should then be *reduced* slightly near the boundary.

5.5 Algorithmic Procedure

The following is a brief summary outlining the procedures necessary for the implementation of the above-described solution-adaptive marching algorithm.

- An initial solution is obtained at the current marching station using the Parabolized Navier-Stokes solver.
- For each coordinate line in a cross-flow direction:
  - The normalized flow property, $\bar{f}$, is computed from selected flow-field variables predicted by the preliminary solution step using Eq. (3.6).
  - An iterative procedure is performed to determine the constant B of Eq. (3.2). At each iteration, Eq (3.2) is used with the current value of B to compute the weighting functions. Equation (3.3) is then used to determine the minimum grid spacing. The iteration procedure terminates when a value of B is found which gives a computed minimum grid equal to the specified minimum grid spacing, Δs_{min} (See Appendix D).
  - From the geometry of Fig. 5.3, the torsion vectors, $\hat{\mathbf{t}}_{c_i}$ and $\hat{\mathbf{t}}_{s_i}$ are computed.
  - The point of intersection of these torsion vectors with the current adaptation line is found.
  - Equation (5.2) is solved for the new arc lengths, s_i, using an iterative procedure. At each iteration, the adaptation boundary procedures are applied as described in the previous section and weighting functions are interpolated to new grid point positions. This iterative procedure terminates when the grid becomes stationary
  - The final grid point positions are interpolated from old values.

- The above steps are repeated for all coordinate lines in the opposite cross-flow direction.
- After all coordinate lines have been adapted in both cross-flow directions, an improved solution is obtained on the resulting refined mesh.
- The refined mesh is proportioned downstream at the next marching station which is used as an initial grid in performing the first step above.

CHAPTER 6

Three-Dimensional Applications: Flow Over Cones

The three-dimensional adaptation methods of the last section are employed in this section for the hypersonic flow over a yawed right-circular cone. The flow-field is computed at several angles of yaw using both the solution-adaptive algorithm and the conventional fixed grid approach. Resulting grids and predicted results are compared to demonstrate the performance of the present algorithm.

6.1 Right-Circular Cone at Incidence

The three-dimensional solution-adaptive PNS algorithm is tested on the hypersonic flow over a 10° right-circular cone at varying angles of incidence, α. The flow-field conditions were chosen to match those of the experimental study conducted by Tracy (1963). These conditions are:

$$M_\infty = 7.95 \qquad \tilde{T}_\infty = 55.4K \qquad \gamma = 1.4$$

$$Re_{\tilde{l}} = 4.14 \times 10^5 \qquad \tilde{T}_w = 309.8K \qquad Pr = 0.72$$

where $\tilde{l} = .1016m$ is the total length of the cone (4 inches). Figure 6.1 illustrates a typical inviscid flow structure for the hypersonic flow over a cone at angle of incidence. Results have been obtained for the flow over a circular cone at angles of incidence, α of 8°, 16° and 24° using both solution-adaptive and fixed grids. For each of these cases, a conical starting solution was obtained at $x = 0.002m$ by employing a stepback procedure and using a fixed conical grid. The conical stepback procedure involves performing an iterative solution procedure on a conical computational slab assuming constant flow properties along rays emanating from a fixed point. After this starting solution was obtained, the solution-adaptive marching procedure was employed to a marching station corresponding to $x = 0.1016m$ for each of the three angle of incidence cases studied.

Table 6.1 provides a summary of the various adaptation parameters used for the multi-directional adaptations at each angle of attack. In this table, Δs_{MAX} and Δs_{MIN} are the maximum and minimum allowable grid spacings and Δs_{ave} is the average grid spacing along the coordinate line. The parameters C_c and C_s are discussed in the previous section (See Eq. (5.4)), and the parameters α and β are constants that multiply the gradient and curvature terms used in determining the weighting functions. The λ is a stiffness parameter assigned to the torsional springs (See Eq. 5.3). Quite frequently, optimum values of these parameters for

adaptation in different cross-flow directions can vary. In Table 6.1 the η- and ζ-subscripts indicate parameters used for grid adaptation sweeps in the respective directions.

Table 6.1: Adaptation parameters.

α	8°	16°	24°
$(\Delta s_{MAX})_\zeta/\Delta s_{ave}$	4.5	4.0	3.2
$(\Delta s_{MIN})_\zeta/\Delta s_{ave}$	.2	.2	.4
$(\Delta s_{MAX})_\eta/\Delta s_{ave}$	2.5	2.0	1.5
$(\Delta s_{MIN})_\eta/\Delta s_{ave}$	.5	.5	.8
λ_ζ	.2	.2	0.5
λ_η	2.	5.	5.
C_s	.5	.5	.5
C_c	.5	.6	.35
α	1.0	1.0	1.0
β	0.0	0.0	0.0

Mach number, pressure and density are the three variables used in the weighting function selection procedure when adapting in the radial direction. For circumferential grid adaptations, the objective was to increase grid resolution on the leeward side of the cone in the circumferential direction. In an attempt to artificially resolve a region of the flow-field known to contain additional complexities, boundary layer thickness was added to this list of adaptation variables when adapting circumferentially. At each circumferential grid point location, the boundary layer thickness adjacent to the grid point was computed and normalized with respect to the maximum value. If this value at a point was greater than the gradients of the above listed variables, then boundary layer thickness was chosen as the weighting function at that point. (i.e. boundary layer thickness was included as the kth value of $|\partial\phi_i/\partial s|_k$ of Eq. 3.6). The maximum value of boundary layer thickness occurs at the leeward meridian for a cone at angle of incidence, while the maximum circumferential pressure gradient occurs closer to the windward meridian. The net effect of using boundary layer thickness when

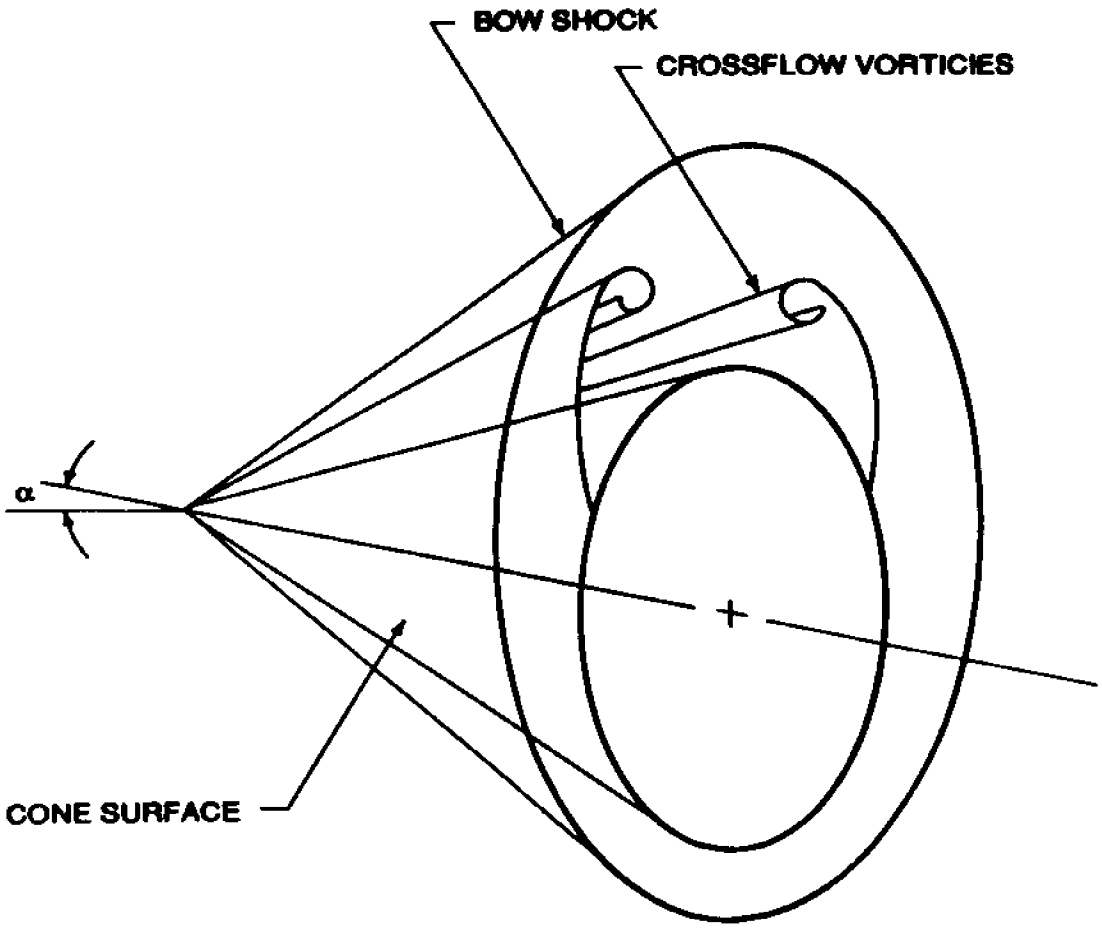


Figure 6.1: Conical flow geometry.

adapting in the circumferential direction was a slight increase in the grid resolution at the leeward side of the cone compared with cases that did not include the boundary layer thickness as an adaptation variable. Additional discussion on the present use of boundary layer thickness to control grid point movement will be provided in the following sections.

Adaptation sweeps were performed in both cross-flow directions at each marching station. The circumferential sweep at each marching station was performed by starting at the cone surface and moving outward. However, the direction of the radial sweeps alternated, sweeping one cross-flow surface from the windward to leeward meridian and the next surface from leeward to windward, etc. These alternating direction sweeps in the radial direction permit information to propagate from both directions, contributing additional smoothness to the grid.

6.2 Grids and Contour Plots

6.2.1 Yaw Angle of 8°: For the 8° angle of incidence case, a 65×65 cross-flow grid was used for both fixed and solution-adapted cases. Figure 6.2 illustrates a sectional view of the resulting solution-adapted computational grid. Grid point clustering is observed at the bow shock location as well as in the boundary layer from the windward to the leeward side of the cone. For this case, the angle of incidence is less than the cone angle; thus, the boundary layer remains attached up to the leeward meridian in the cross-flow plane. However, the boundary layer thickens as the flow moves toward the leeward meridian in the cross-flow direction, and, as this thickening progresses, the grid adaptation supplies the boundary layer with additional grid points. Figure 6.3 shows a comparison between fixed grid and solution-adapted grids at $x/\bar{l} = 1$.

Figure 6.4 shows contours of constant Mach number at this same x-location. Improvements in bow shock resolution are observed in the solution obtained using

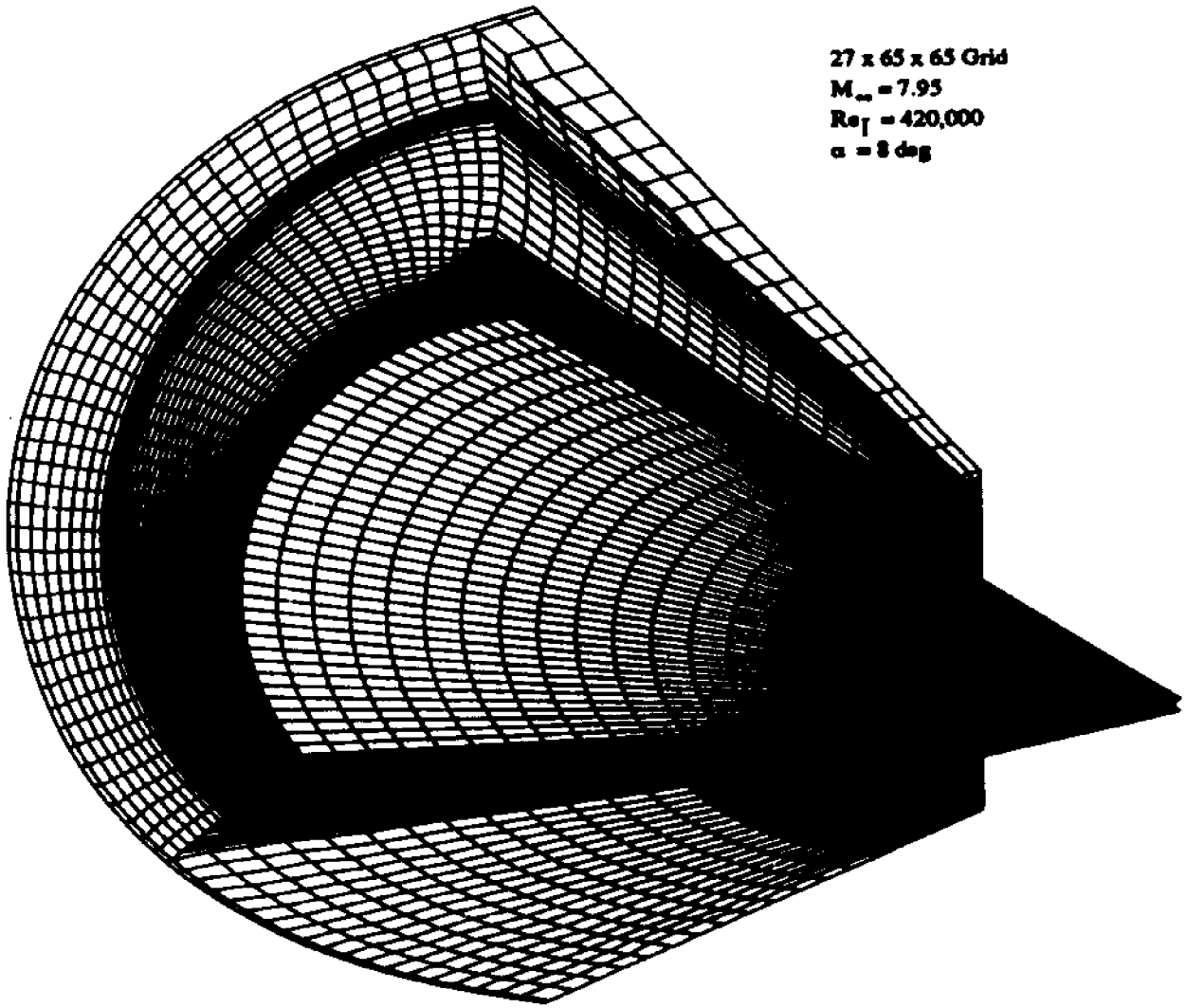


Figure 6.2: Sectional view of solution-adapted computational grid.

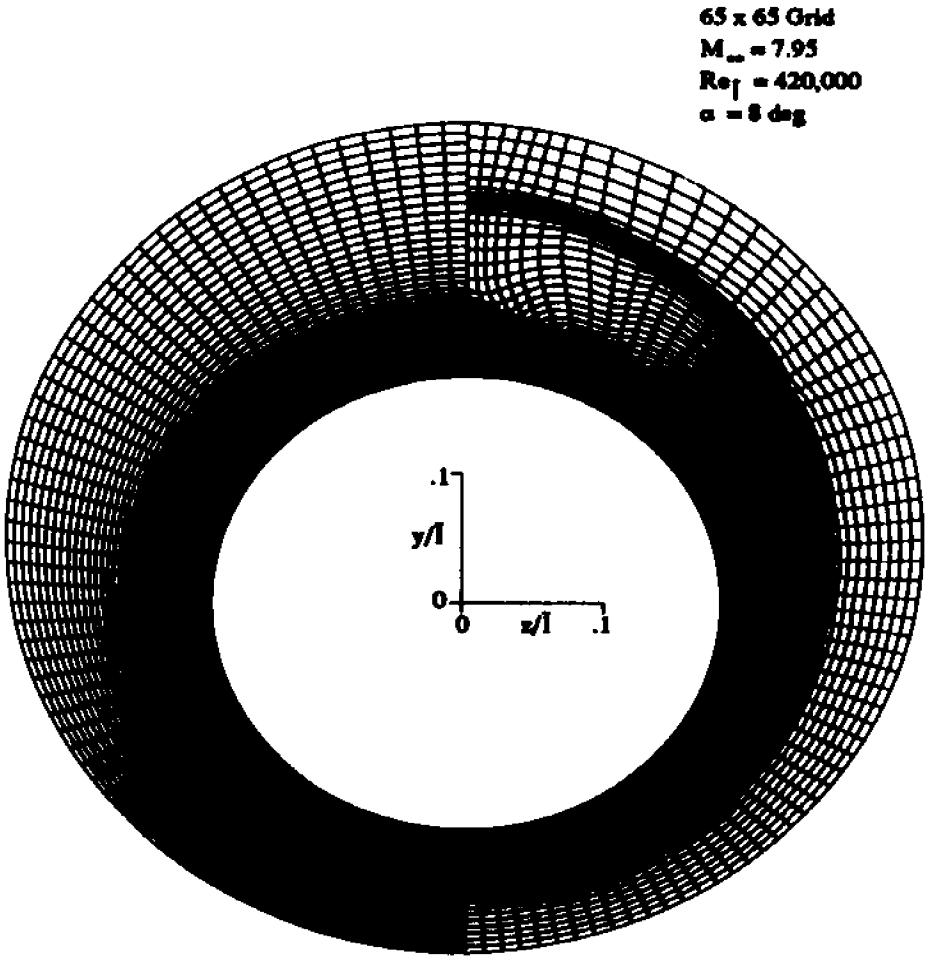


Figure 6.3: Comparison of crossflow computational grids at x= 0.1016m. Left-half, fixed grid; right-half, solution-adapted grid.

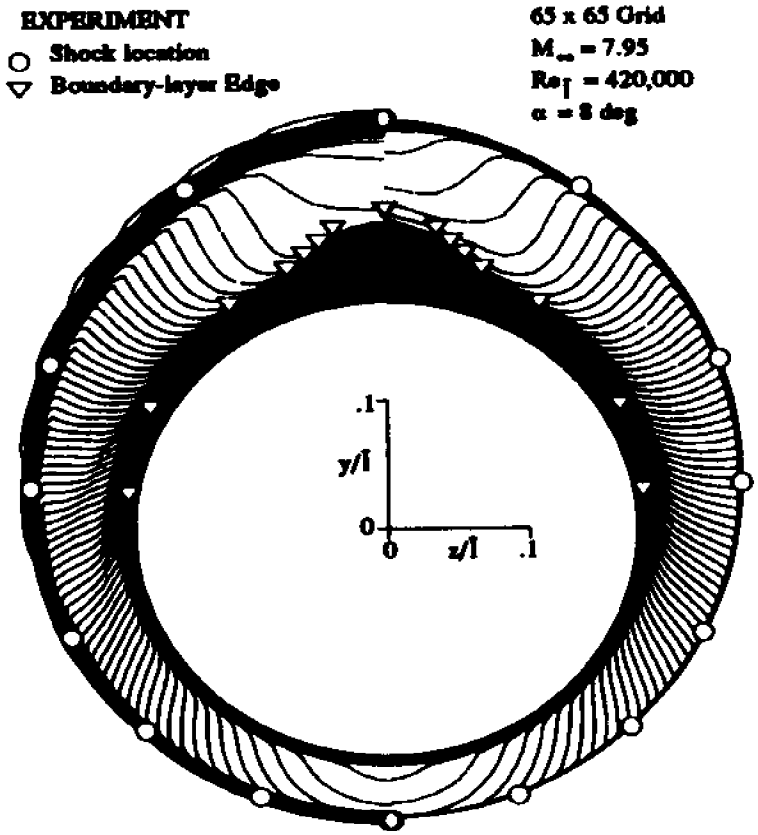


Figure 6.4: Comparison of Mach number contours at x= 0.1016m. Left-half, fixed grid; right-half, solution-adapted grid.

the adapted grid over that of the fixed grid. The inverted triangle symbols of this figure indicate the boundary layer edge deduced experimentally by Tracy while the circles indicate experimentally determined shock wave location. The locations of these features, as observed by Tracy[14] are well predicted by the solution-adaptive PNS solver.

6.2.2 Yaw Angle of 16°: A 70×70 cross-flow grid was employed for both fixed and adapted cases at a 16° angle of yaw.

Figure 6.5 shows a sectional view of the resulting solution-adapted grid for this case. This figure illustrates the true three-dimensional nature of the present problem. Each cross-flow plane has been adapted in both cross-flow directions. The torsional influence in the streamwise direction has prevented relative skewness between adjacent cross-flow planes permitting the marching process to proceed without interruption. The position of the bow shock surface is marked by the outermost grid clustering which extends around the entire figure. Also evident in this figure is a gradual thickening of the clustering region near the cone surface in the circumferential direction illustrating the cross-stream structure of the boundary layer. Its point of separation from the cone surface is marked by the breakup of this grid clustering region. The grid points remain clustered in the detached viscous layer until a second division of the clustering region occurs where the flow rolls up into a vortical-like structure.

For the 16° angle of incidence case, the radial adaptation process at each marching station was performed by sweeping grid lines from the windward meridian to the leeward meridian. On the other hand, for the 8° angle of incidence case the direction of radial adaptation sweeps was alternated at each successive marching station as described above. The absence of alternating direction adaptations can cause a slight waviness to appear in the grid as seen in the leeward symmetry plane of Fig. 6.5. This waviness is not present in the symmetry planes

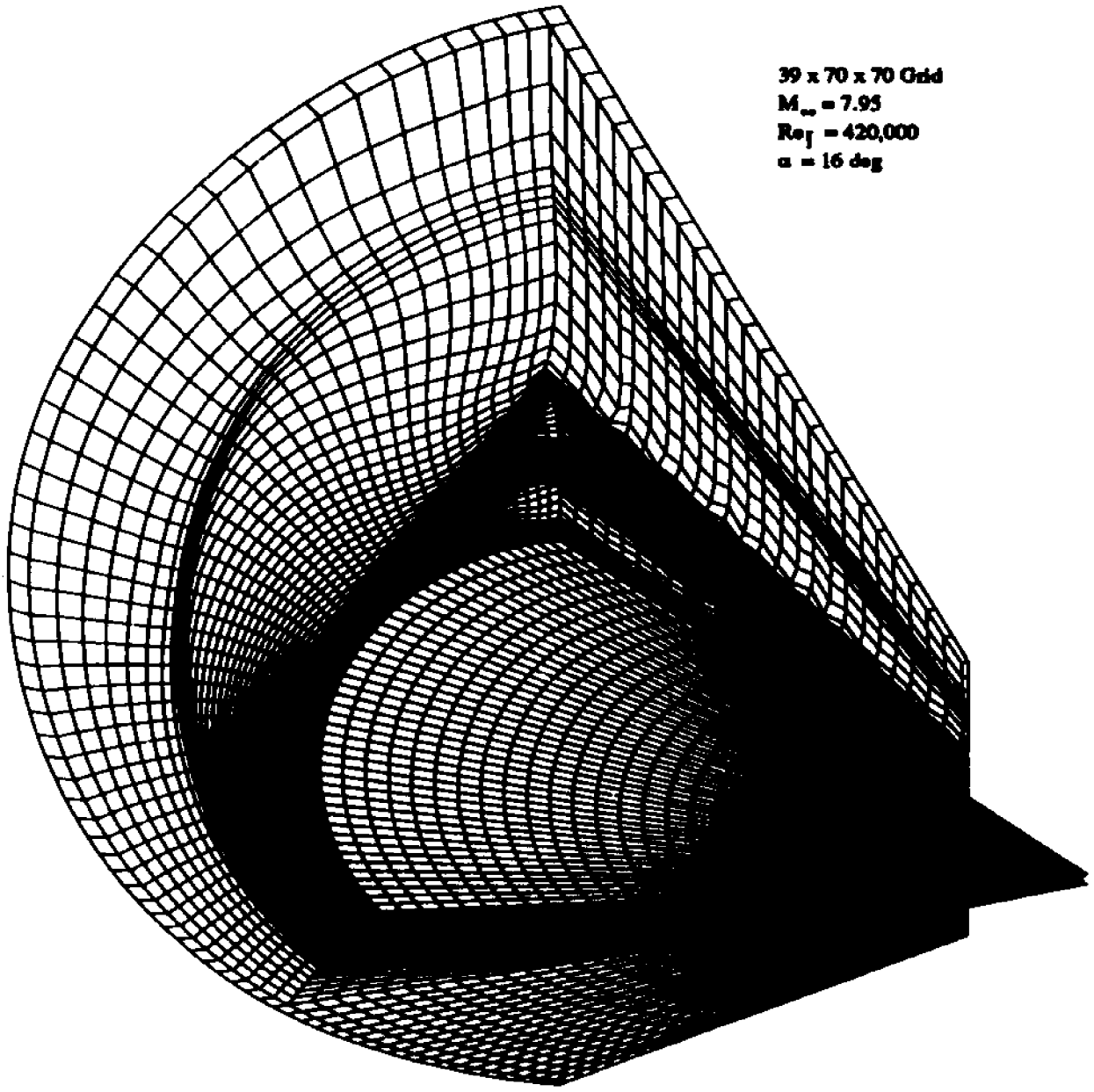


Figure 6.5: Sectional view of solution-adapted computational grid.

of the grid of Fig. 6.2 where solution-adaptation was preformed using alternating direction sweeping technique.

When the angle of incidence is increased from 8° to 16,° it becomes greater than the cone half-angle and, as a result, a region of cross-flow separation develops on the leeward side of the cone where the viscous layer detaches from the cone surface. This separation results from the increasingly adverse cross-flow pressure gradient on the leeward side of the cone. A rotational feature in the flow-field is evident in the solution-adapted mesh shown on the right half of Fig. 6.6. Increased grid clustering is observed around the bow shock region and detached shear layer. The clustering region on the leeward side of the cone and situated about halfway between the top of the detached shear layer and the cone surface is due to the circumferential grid lines attempting to roll up and imitate the shape of the existing cross-flow vortical structure.

The experiments of Tracy detect weak embedded cross-flow shock waves at the onset of the separation region where the Mach number due to the cross-stream velocity components exceeds unity. The experimentally observed positions of these embedded shocks, in addition to the bow shock position, are illustrated by the overlaid circles of Fig. 6.7. Again, excellent agreement in the experimentally observed position and the computed position of the bow shock is observed for both the fixed and adapted grid solutions. The Mach contours of Fig. 6.7 also show improved resolution of the bow shock for the solution-adapted case over the fixed grid case. However, the embedded shocks are not evident in the Mach contours of this figure due probably to their relative weak strength. The inverted triangle and triangle symbols of Fig. 6.7 show the locations of the outer and inner edges, respectively, of the detached viscous layer. This experimental data was deduced from the pitot pressure measurements of Tracy.[14]

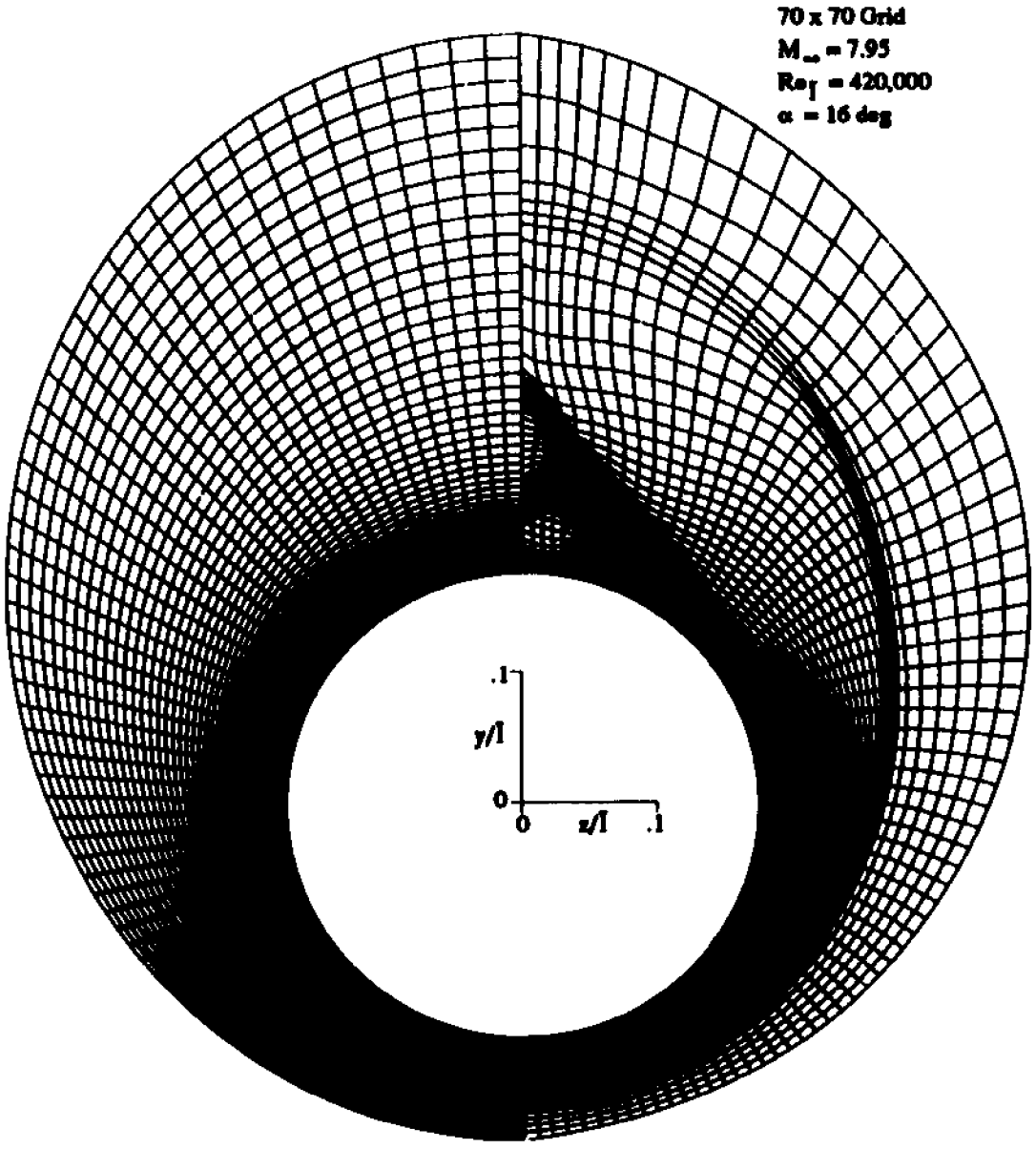


Figure 6.6: Comparison of crossflow computational grids at x= 0.1016m. Left-half, fixed grid; right-half, solution-adapted grid.

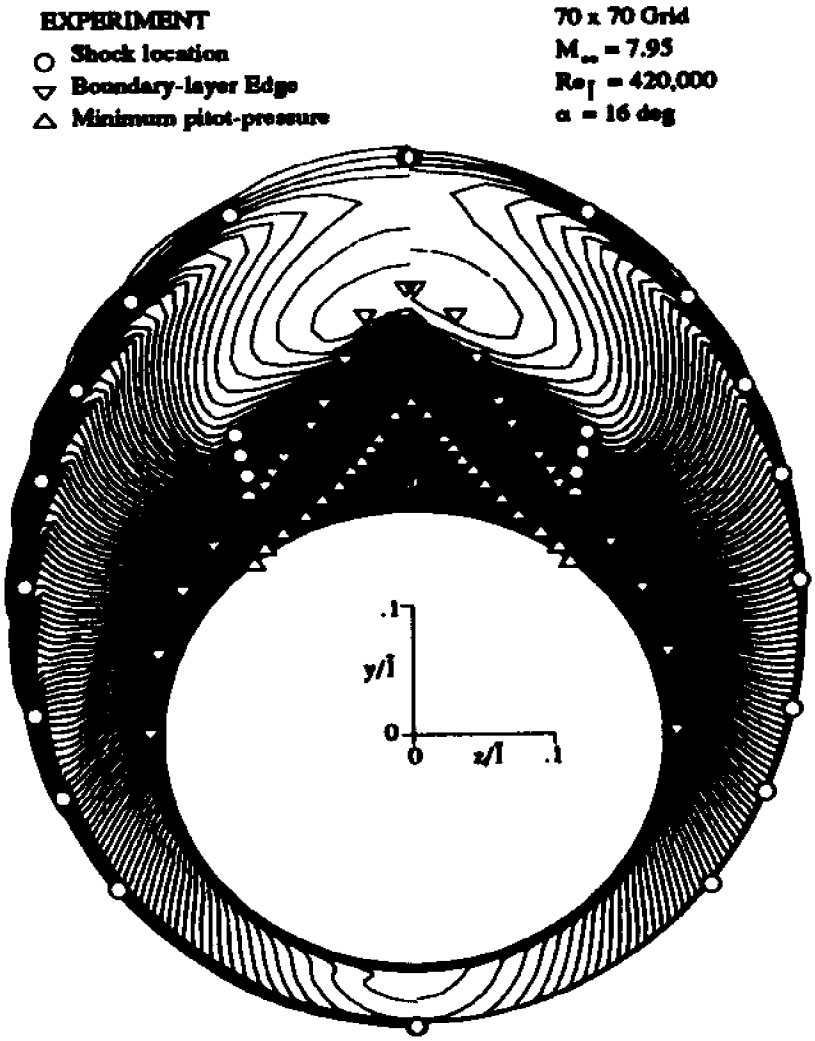


Figure 6.7: Comparison of Mach number contours at x= 0.1016m. Left-half, fixed grid; right-half, solution-adapted grid.

6.2.3 Yaw Angle of 24°: When the angle of incidence is further increased to 24°, additional thickening of the separated region occurs at the leeward side of the cone. The viscous layer starts out running along the cone surface in the circumferential (cross-flow) direction. Close to the leeward side it separates from the surface and turns sharply upward becoming vertical, running parallel to the leeward symmetry plane and away from the cone surface. The detached viscous layer then turns sharply back towards the symmetry plane, intersecting normal to it. Because of the abrupt changes in the direction of this shear layer, the adaptive grid algorithm has to choose which family of coordinate lines is best suited to resolve this shear layer while simultaneously maintaining a sufficient degree of grid orthogonality.

Figure 6.8 shows a grid comparison for the fixed (left-half) and solution-adapted (right half) cases, each side consisting of a 90×90 grid. Grid point clustering is observed where the viscous layer detaches from the cone surface and turns upward. Before this change in direction, the viscous layer is running essentially parallel to the circumferential family of grid lines and can easily be resolved by them. However, following its detachment, the viscous layer turns almost normal to circumferential lines, making it difficult for them to adjust and, due to orthogonality restrictions, disallows any significant resolution from coordinate lines of the opposite family. This lifting of the circumferential lines also causes the radial lines to bend inward, away from the symmetry plane, leaving less grid resolution in this region.

Mach number contours are plotted in Fig. 6.9 for both the fixed grid (left-half) and adapted (right-half) cases at a 24° yaw angle. The overlaid circles represent shock wave positions as observed experimentally by Tracy.[14] The inverted triangles and triangles represent the outer and inner edges, respectively, of the shear layer as deduced from the pitot pressure measurements of Tracy. An increase in resolution of the top of the shear layer can be seen in the adapted side

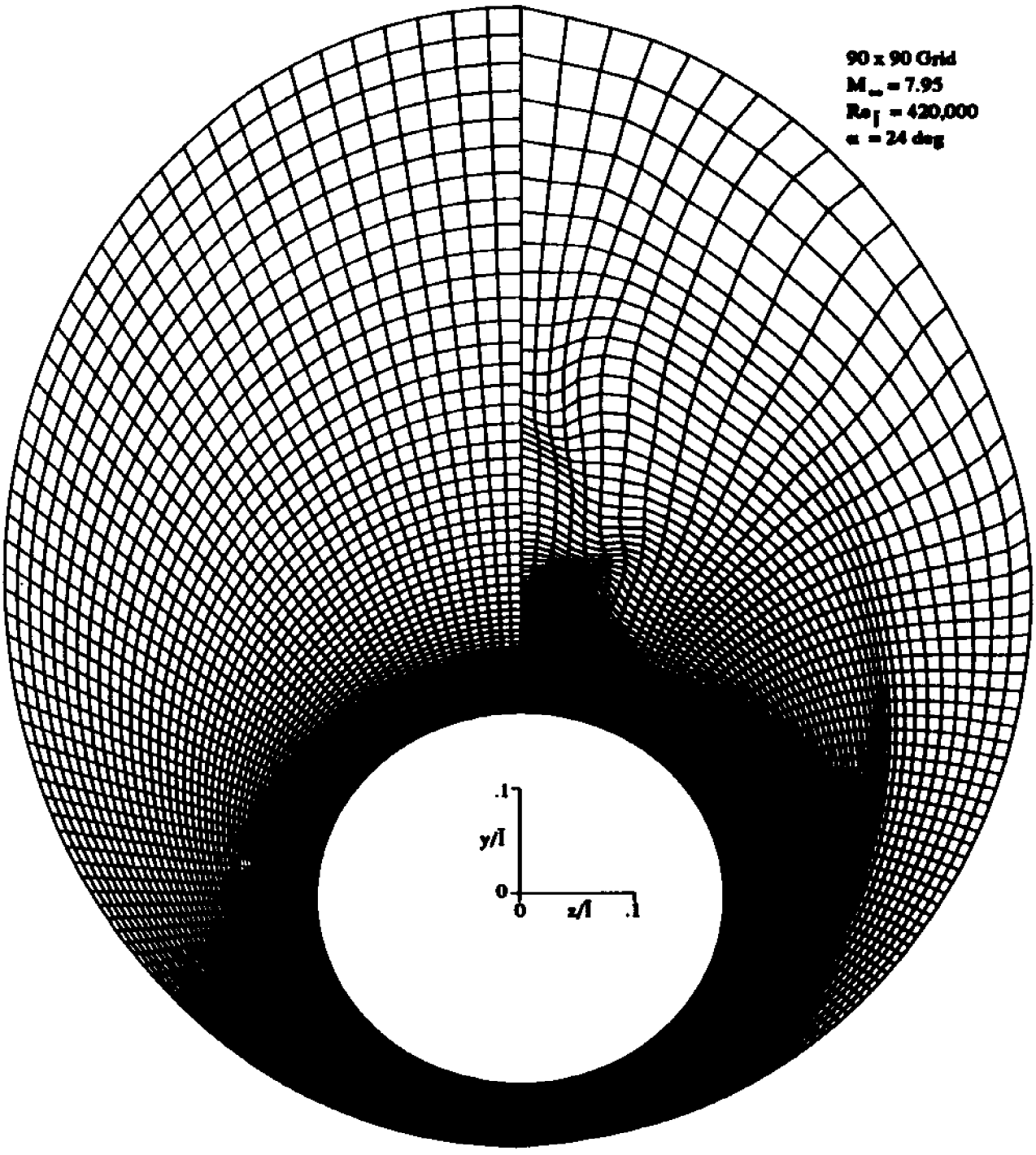


Figure 6.8: Comparison of crossflow computational grids at x= 0.1016m. Left-half, fixed grid; right-half, solution-adapted grid.

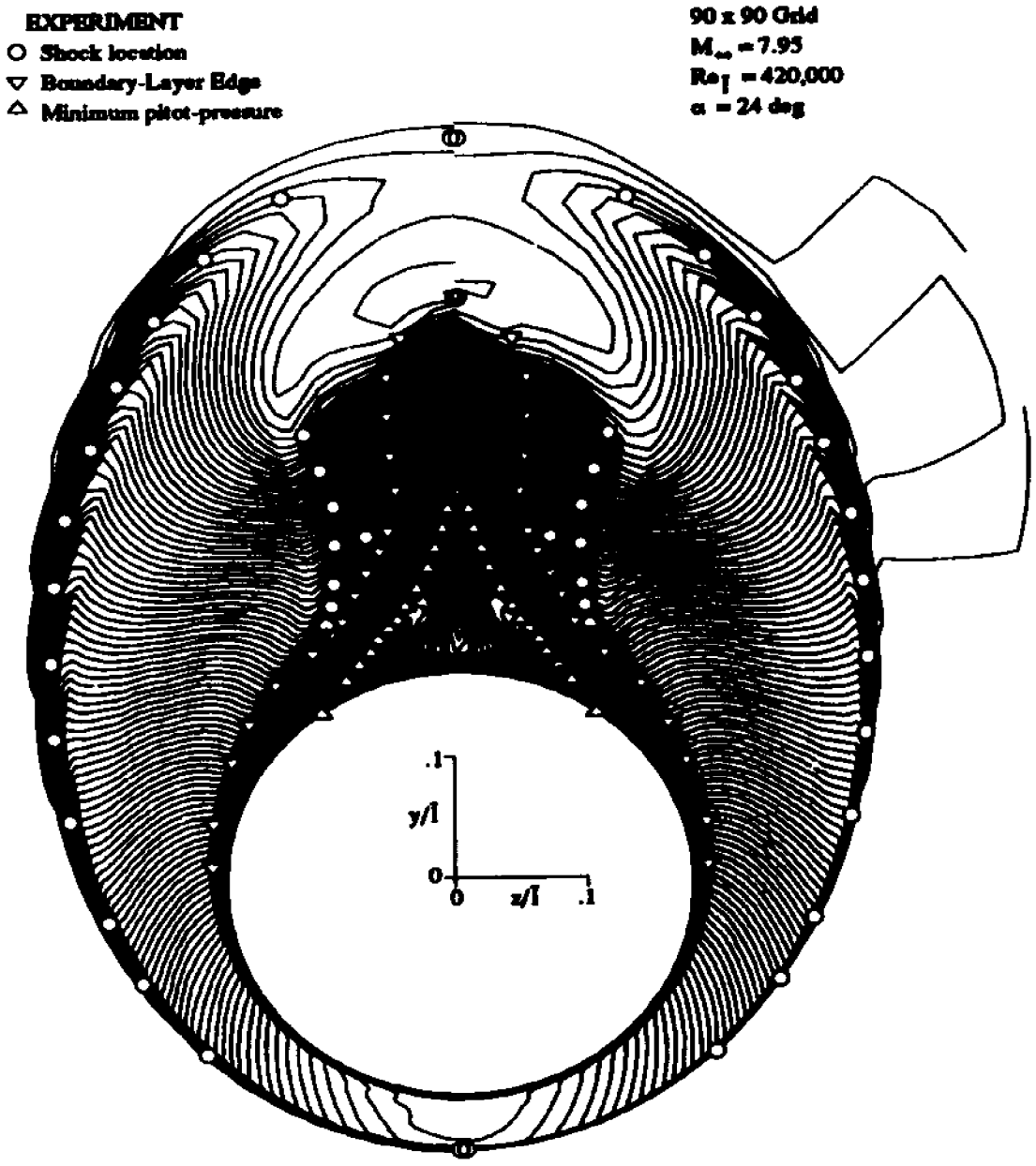


Figure 6.9: Comparison of Mach number contours at x= 0.1016m.
Left-half, fixed grid; right-half, solution-adapted grid.

of this figure although this increase in resolution is less apparent for the portion of the shear layer that runs almost vertically upwards. This loss of resolution is due to rather strict orthogonality constraints imposed on the grid.

In an attempt to resolve as much of the detached shear layer on the leeward side of the cone as possible, solution-adaptive calculations were performed by adapting the grid in each cross-flow direction separately. The resulting grid for the radial adaptations is illustrated by the left-half of Fig. 6.10 while the right-half resulted from adapting in the circumferential direction only. As in Fig. 6.8, both halves of this figure consist of a 90×90 computational grid. Figure 6.10 illustrates how each coordinate line family resolves the detached shear layer while simultaneously attempting to maintain a minimum degree of grid orthogonality. For the radial adaptations (left-half of Fig. 6.10), grid lines on the leeward side of the cone must first turn sharply away from the cone surface and then turn back normal to the leeward meridian in order to resolve the detached shear layer all the way to the top. Some of the grid orthogonality is lost due to these sharply turning circumferential lines of the left-half of Fig. 6.10, more so in the region where the detached shear layer becomes almost normal to the cone surface. The radial adaptations alone (left-half of Fig. 6.10) did very well in clustering grid points in the bow shock location. The resulting cross-flow grid is accurately aligned with the bow shock the entire length of the grid, from windward to leeward meridians. This increase in alignment and resolution throughout the grid of Fig 6.10 (left side) resulted from performing the radial adaptations using the alternating direction sweeping technique which allows for information to propagate from both sides of the grid (windward and leeward meridians).

About halfway between the top of the shear layer and the cone surface at the leeward meridian, another grid clustering region exists inside the detached shear layer. This same type of grid structure was also observed in Figs. 6.6 and 6.8.

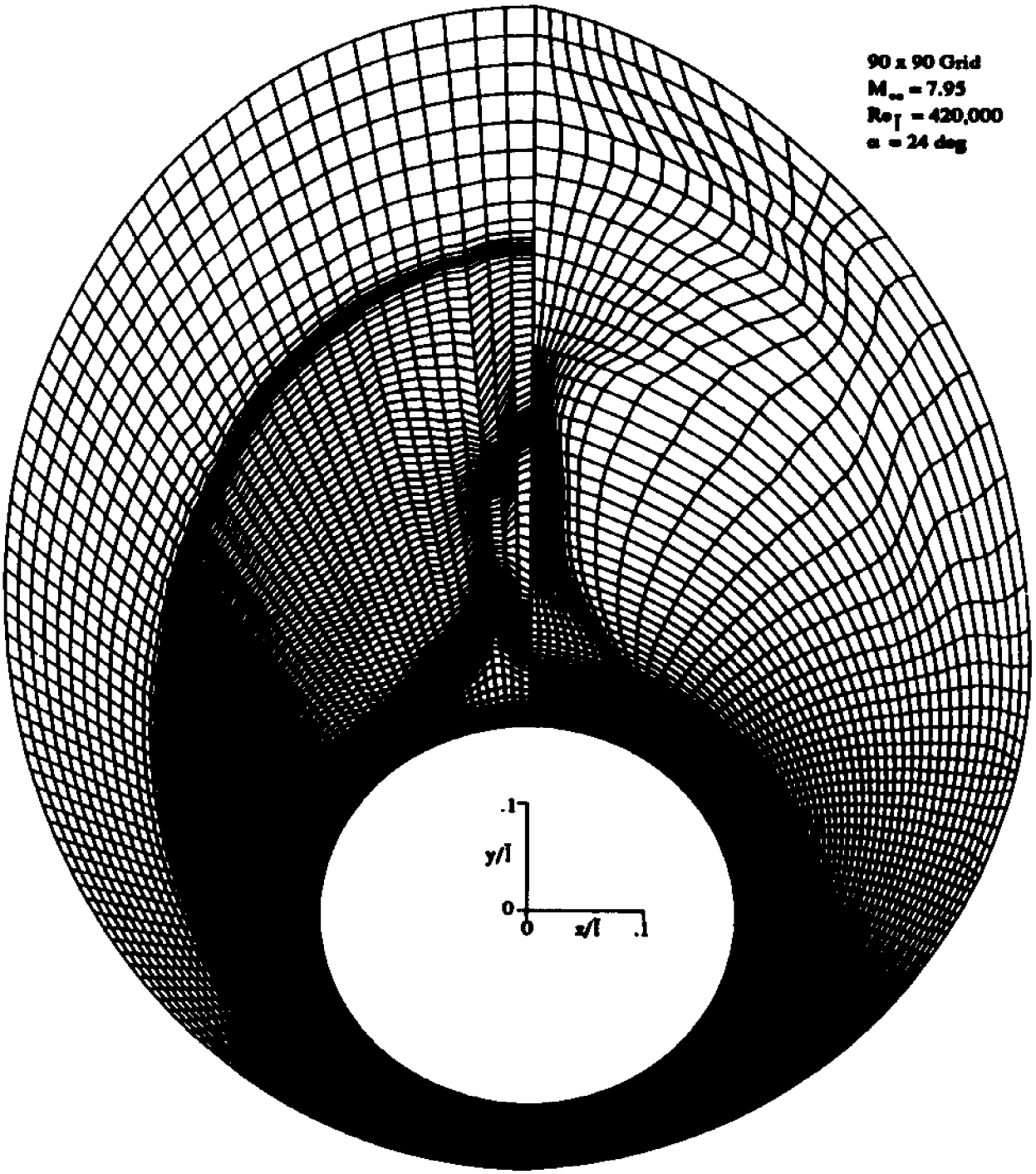


Figure 6.10: Comparison of crossflow computational grids at x= 0.1016m. Left-half, adapted in radial direction only; right-half, adapted in circumferential direction only.

The circumferential only adaptations in the right-half of Fig. 6.10 illustrate improvements in circumferential grid resolution at the leeward symmetry plane over the multi-directional adaptations of Fig 6.8. The reason for this behavior can best be explained with a discussion on the manner in which flow-field gradients used in constructing the weighting functions are computed. The derivatives are computed in the direction of the grid line. In the present test cases, as a result of the radial grid adaptation sweep, circumferential grid lines tend to align themselves with the detached shear layer at the leeward side of the cone. Consequently, when the circumferential adaptation sweep is performed, the weighting function is computed as a derivative in the direction of the grid line (no longer in a strict circumferential orientation), which runs parallel with the shear layer. For the circumferential only adaptations of Fig. 6.10, the positions of the circumferential grid lines were not disturbed by radial adaptations and, unlike the multi-dimensional adaptation cases, cut directly through the shear layer, in a direction of strong flow-field gradient. When adapting in both directions simultaneously, the circumferential grid lines become parallel with the detached shear layer, along which there are no strong gradients to use as weighting functions. This is another reason the weighting function was modified with boundary layer thickness for circumferential adaptations when adapting in both cross-flow directions simultaneously. For each half of Fig. 6.10, the grid can be seen to cluster around the separated shear layer all the way to the top, as desired.

Figure 6.11 shows Mach contours overlaid with the symbols as in the previous angle of incidence cases. The left-half of this figure shows improvements in bow shock resolution over all the 24° angle of incidence cases presented thus far. The detached shear layer in both halves of Fig 6.11 (uni-directional adaptations only) is more clearly resolved than in either half of Fig. 6.9 (fixed grid or multi-directional adaptation cases). The increased circumferential resolution provided

by the circumferential only adaptations (right-half of Fig. 6.11) allows for a more clearly defined viscous layer which extends slightly higher than the viscous layer predicted with the fixed grid. This same degree of resolution is also observed using radial adaptations alone (left-half of Fig. 6.11), although a slight waviness exists in the vertical portion of the shear layer due to the lack of grid orthogonality in this region. The inner portion of the viscous layer that was predicted with either of the uni-directional adaptations of Fig. 6.11 is more clearly defined than that predicted using either the fixed grid or the multi-directional adaptations of Fig. 6.9.

No significant improvement can be detected in the resolution of the embedded shock waves, even with the use of the adaptive grid. This is due primarily to an insufficient number of radial grid lines available to cluster in these regions. For circumferential adaptations alone, all available grid points move towards the shear layer, leaving less grid resolution in embedded shock regions. These embedded shock waves as observed by Tracy are indicated by the overlaid circles emanating from the detached portion of the shear layer in a lambda shaped pattern.

6.3 Pressure and Heat Transfer Comparisons

Figure 6.12 shows a comparison of pitot pressure profiles in the leeward meridian plane for all three angle of incidence cases using both solution-adapted and fixed grids with experimental results obtained by Tracy. The results for the solution-adapted cases show improvements over the fixed grid cases at all angles of incidence.

The experiments show an abrupt increase in the pitot pressure at the inner edge of the detached shear layer. The solution-adapted 24° angle of incidence cases predict this feature in the shape of the pitot pressure profile fairly well. The 24° fixed grid solution predicts a more gradual rise in pitot pressure through the detached shear layer. The solution-adaptive PNS solver under-predicts the

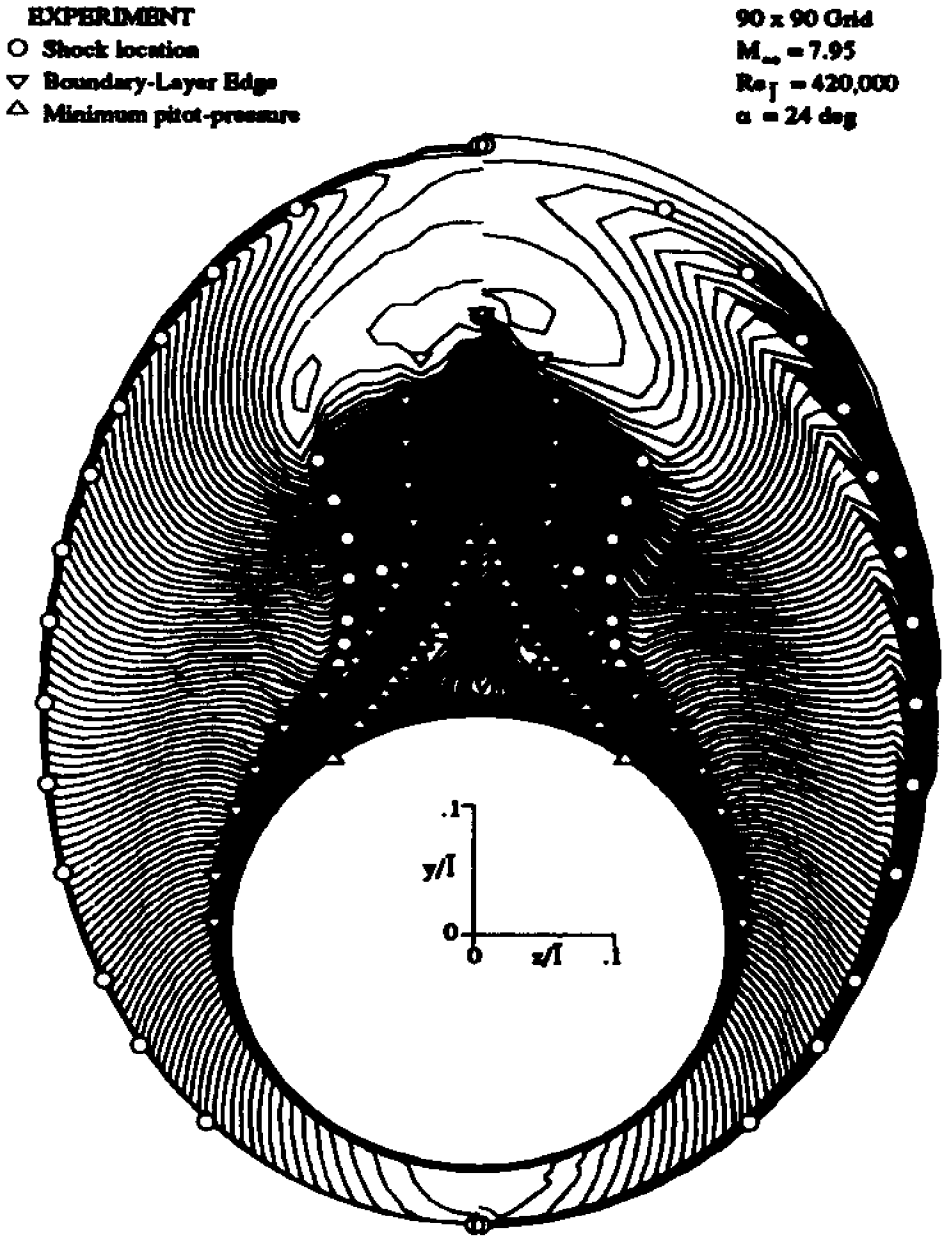


Figure 6.11: Comparison of Mach number contours at x= 0.1016m. Left-half, adapted in radial direction only; right-half, adapted in circumferential direction only.

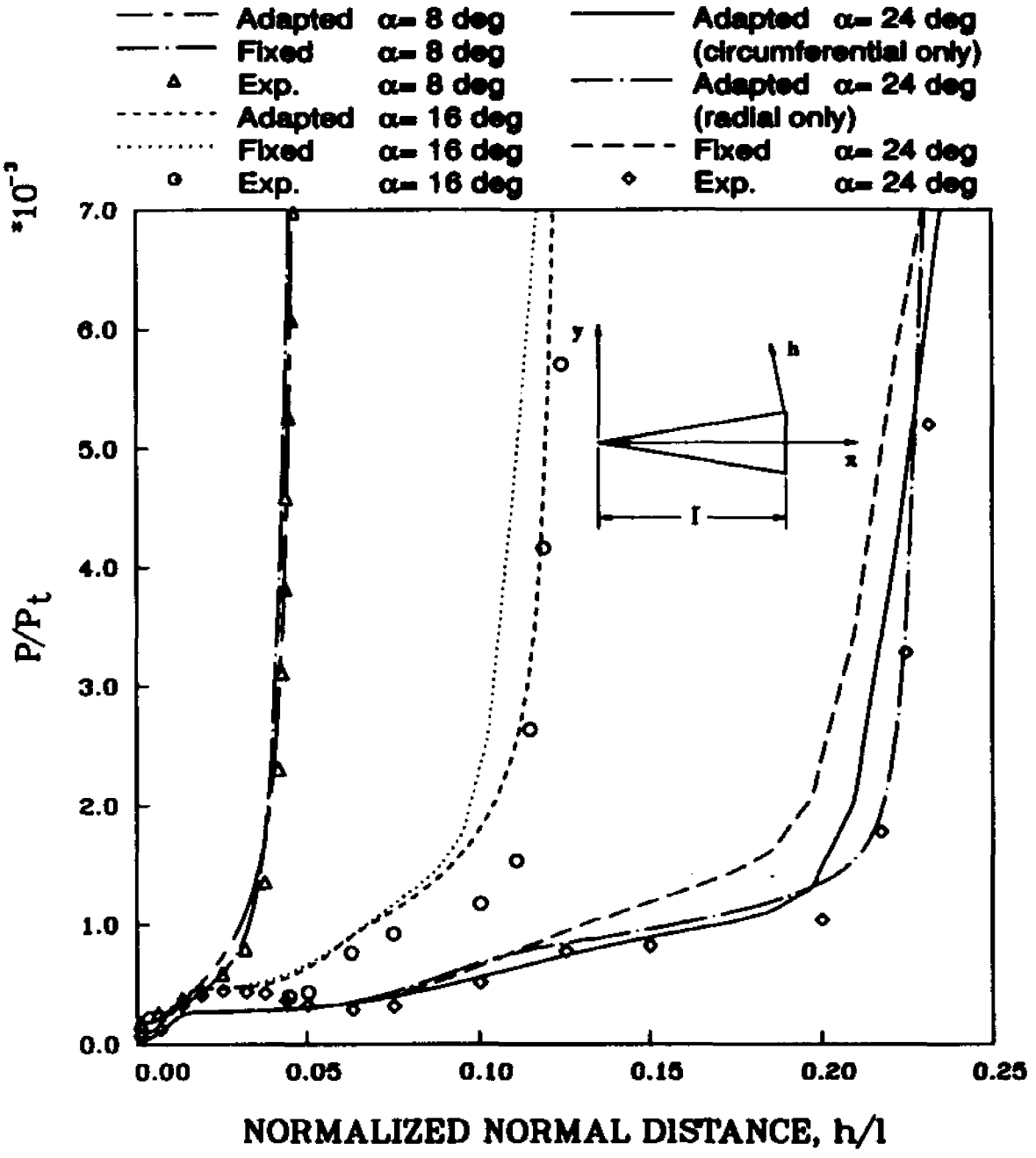


Figure 6.12: Comparison of pitot pressure profiles at leeward meridian.

slight bulge in pitot pressure which exists near the cone surface. One possibility which may improve the prediction of this feature is to adapt the points on the surface of the cone, thus allowing for a finer circumferential grid resolution near the cone surface. In all cases presented for this work, the surface points were not permitted to move. Due to orthogonality constraints at the surface, the movement of grid points a small distance above the surface was limited, disallowing further circumferential grid refinement. Pitot pressure predictions for the 24° cone case using multi-directional adaptation sweeps were found to be only slightly better than those predicted using the fixed grid and are not included in Fig 6.12.

Figure 6.13 compares circumferential pressure distributions with those obtained experimentally by Tracy at a marching station corresponding to $x = 0.1016m$. The results computed with the solution-adaptive PNS solver agree very well with experiment; agreement is not significantly influenced by the adaptation of the grid.

Figure 6.14 compares Mach number profiles at the leeward meridian in the bow shock region for each cone case. Significant improvements can be seen in the resolution of the bow shock for the 8° and 24° (radial only) adapted cases. Improvements in shock definition obtained using the adaptive grid are not as pronounced for the 16° angle of incidence case due to the lack of grid resolution in this region (See Fig. 6.6) This is due primarily to the lack of alternating direction sweeps for this case as discussed in a previous section. In adapting the grid for the 16° cone case, the radial adaptations were performed by sweeping lines from windward to leeward only. For the 8° and 24° (radial only) cases, the alternating technique discussed earlier was employed, allowing information to propagate from both leeward and windward sides of the cone.

Figure 6.15 compares computed heat transfer rates normalized by the heat transfer rate at zero incidence. These results agree well with experiment.

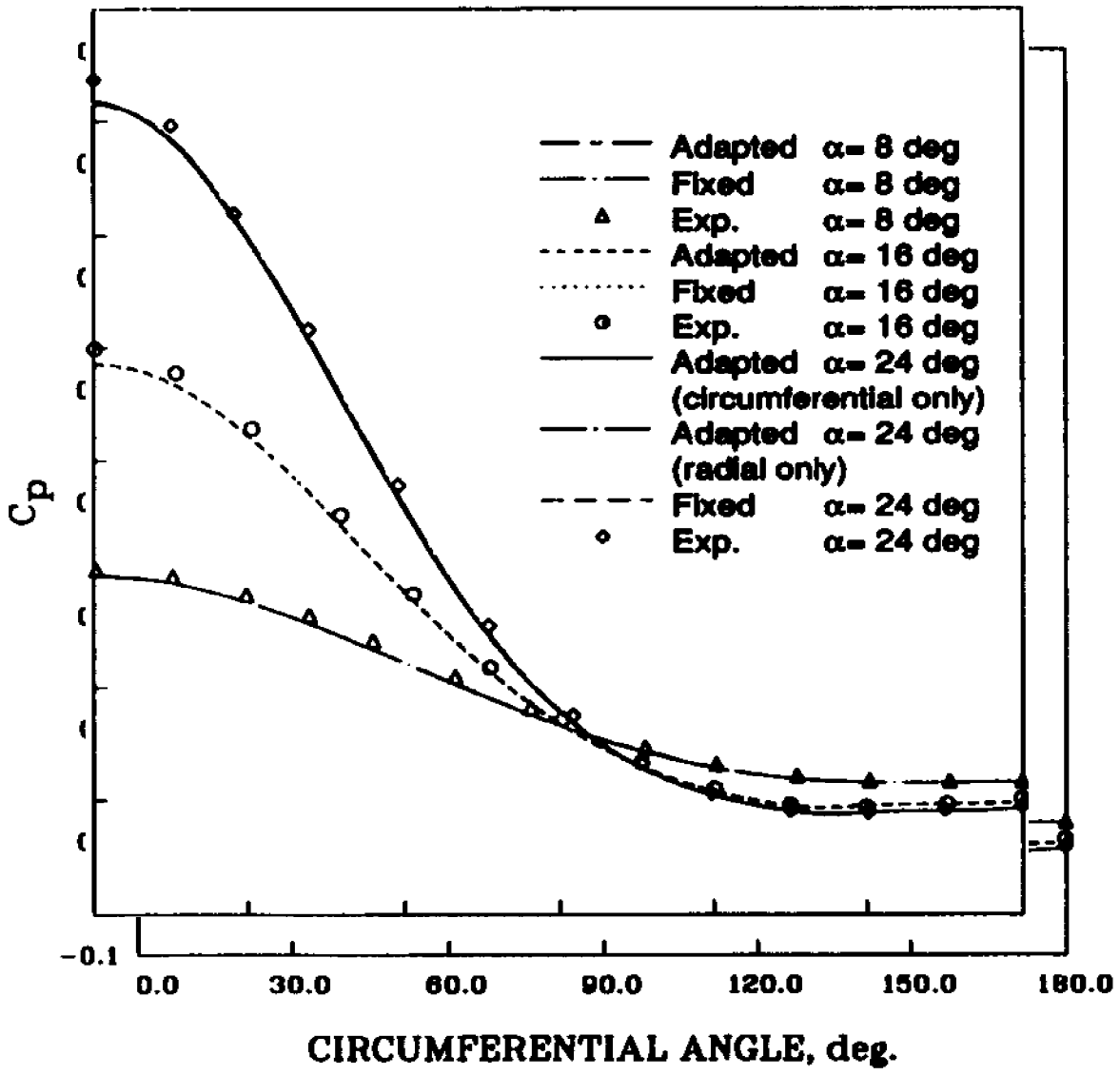


Figure 6.13: Circumferential surface pressure distributions at x= .1016m.

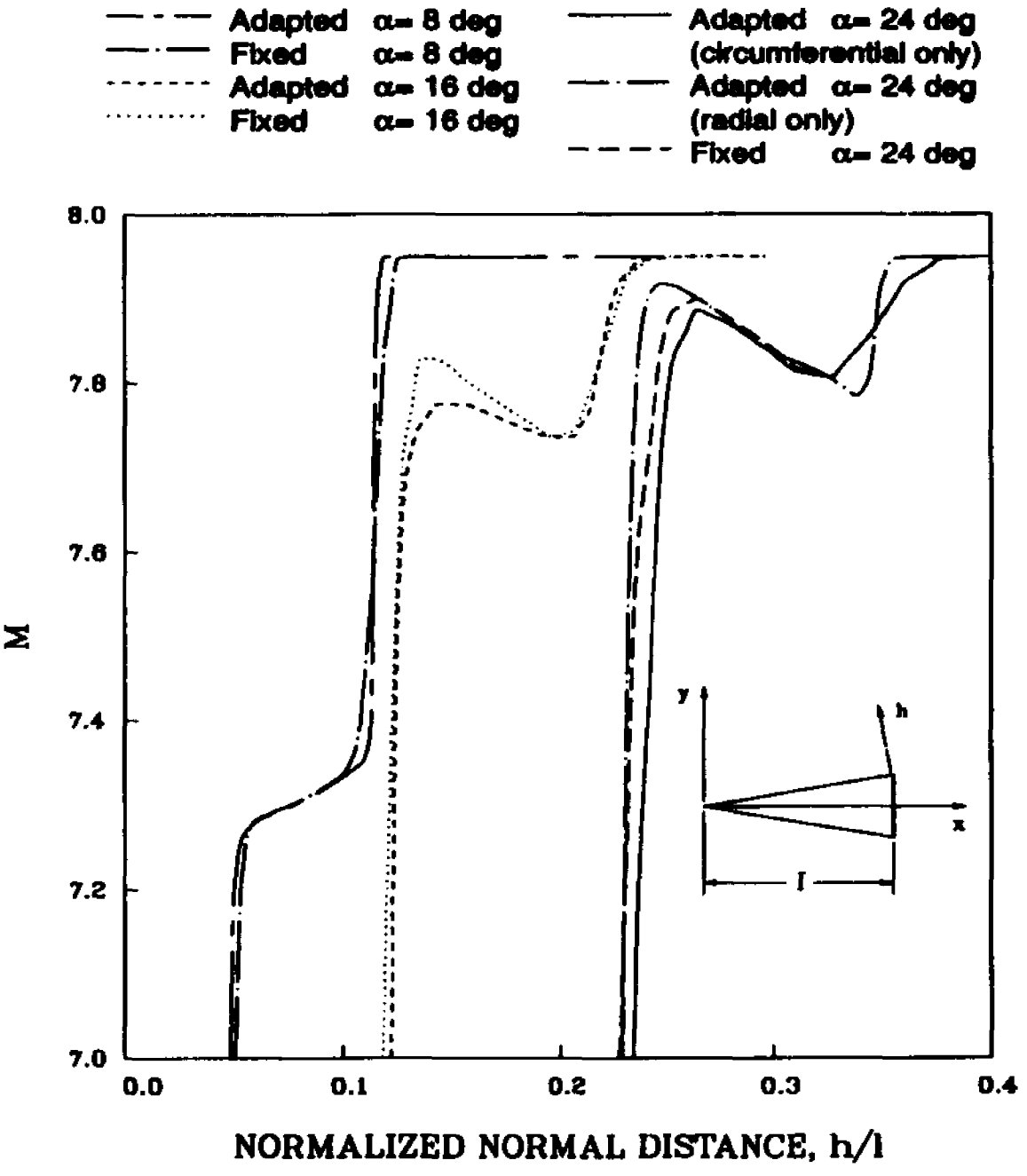


Figure 6.14: Comparison of Mach number profiles at leeward meridian.

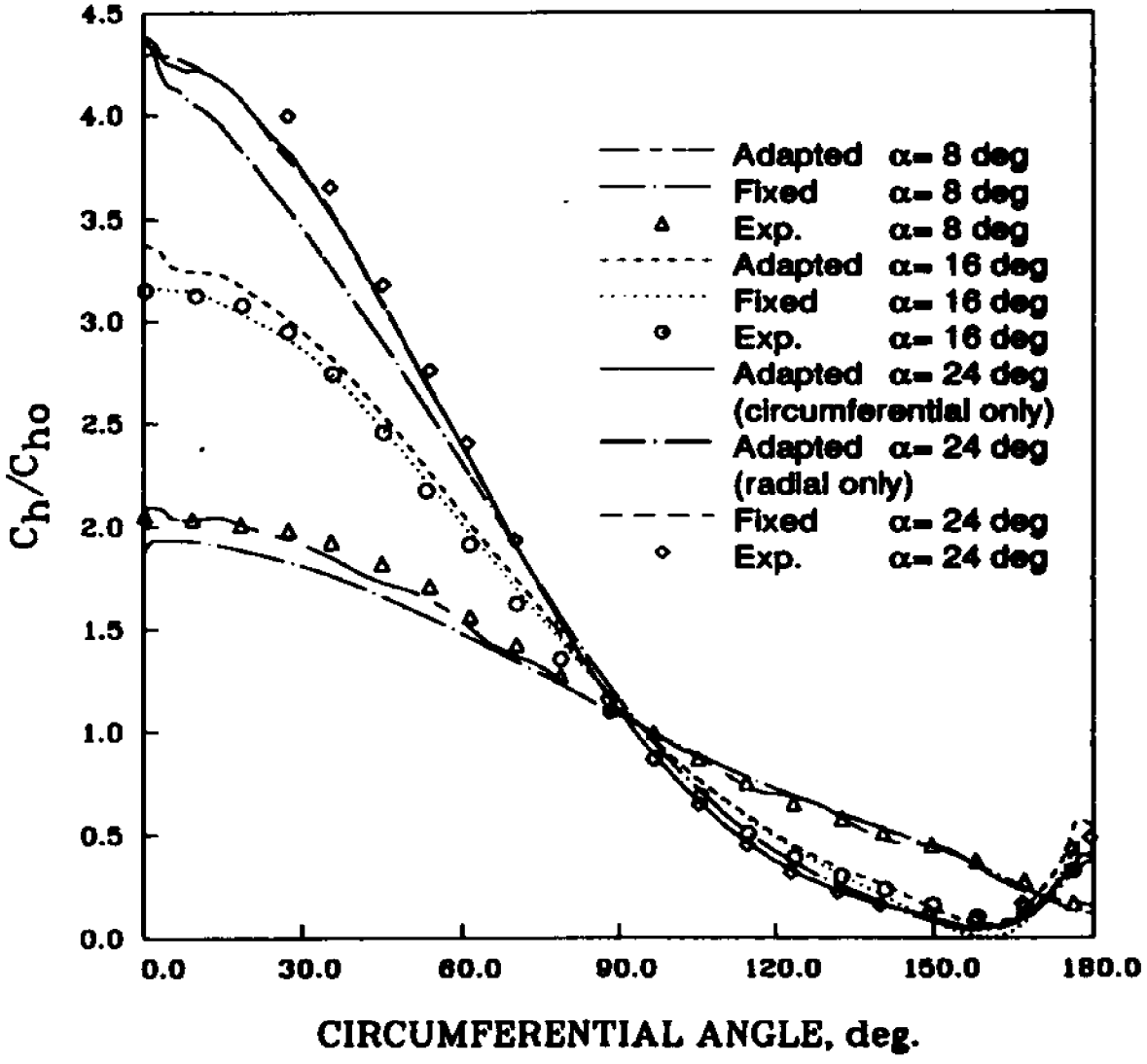


Figure 6.15: Circumferential heat transfer distributions at x= .1016m.

CHAPTER 7

Application to Complex Geometries

The ideas presented in the preceding chapters are now extended to more complex three-dimensional configurations. In addition, this chapter also presents the development of a three-dimensional grid-fitting algorithm similar, in concept, to the two-dimensional grid-fitting procedure described earlier. This algorithm provides a means of identifying the position of the bow shock (or outermost shock) and adjusting the extent of the grid in order to minimize the number of grid points in the freestream. In this manner, the grid is adjusted at each marching step, maintaining a maximum grid density inside the region of solution variation.

The procedure for incorporating the torsional terms from the streamwise and cross-flow directions has been improved in the present chapter. Instead of specifying two torsional terms, one each in the streamwise and cross-flow directions, a single torsional term is used. This eliminates the problem of conflicting grid reference positions that is present in the method used in the preceding chapter.

7.1 Algorithm Improvements

In a typical space-marching algorithm the grid plane at the current marching station and the grid plane at the next marching station a distance $\Delta\xi$ downstream make up a computational slab. When performing the space-marching solution procedure, it is necessary to be able to define the cross-flow geometry of the body at any arbitrary streamwise location. However, a typical surface grid is composed of a finite number of specified cross-flow surface sections (ξ-intervals), and the marching stepsize, $\Delta\xi$, generally differs appreciably from the size of these surface grid ξ-intervals. Therefore, cross-flow grid planes are first constructed at locations corresponding to each ξ-interval on the input surface grid. This can be done during the marching process, one-by-one, as each new ξ-interval is encountered using conventional grid generation techniques The grid planes at the actual marching stations, which are grid planes intermediate to the grid planes at the surface ξ-intervals, can then be obtained by linear interpolation between the grid planes on either side.

In the marching solution-adaptive grid procedure developed in the preceding sections, the grid at each subsequent marching station downstream of the current marching station is obtained by linear interpolation between the adapted grid plane at the current marching station and the grid plane at the next ξ-interval of the surface grid downstream. The latter grid plane is updated after adaptation of the current marching station by performing a sequence of one dimensional interpolations along each coordinate line separately. Thus, the two planes between which a preliminary cross-flow plane is interpolated are sufficiently aligned with each other so that the linear interpolation produces a grid plane at the next marching station that is aligned with the current marching station. In this manner, the complex problem of three-dimensional grid point interpolation is avoided.

This method of aligning the upstream and downstream sides of the computational slab in preparation for numerical integration was, however, found to be restricted to relatively simple bodies whose surface geometry changes at some constant rate (such as the right circular cone of the preceding chapter) and is not suited to more complex configurations such as those contained in the present work.

In as much as the flow properties are not forewarned of the changing geometry downstream of current marching station, neither is the adapted grid. Once the grid plane at the current marching station is adapted, the task arises of building a grid plane at the next marching station. These two adjacent grid planes must be sufficiently aligned with one another in order to make up a computational slab which will permit accurate numerical integration of the flow equations. As the solution-adaptive marching scheme proceeds downstream, the grid becomes more aligned with the flow-field structure. As the marching planes become more adapted, and if the surface geometry changes appreciably in the marching direction, the task of constructing an adjacent grid plane at the next marching station a distance $\Delta\xi$ downstream of the newly adapted marching station can become increasingly complex. In fact, this is the major obstacle in implementing an effective three-dimensional solution-adaptive marching procedure.

In this section, the cross-flow grid at each marching station is obtained from a downstream projection of the grid at the previous marching station. Instead of trying to align the downstream grid plane with the adapted grid plane and then performing an interpolation between these two grid planes to construct the computational slab (as was done for the cones of the last chapter), it is found to be a more general procedure (applicable to more general three-dimensional geometries) to project the adapted grid plane at the current marching station downstream to the next marching station using the fixed surface grid as a guide.

The only additional information needed (in addition to a complete surface grid) is that concerning how far out into the freestream the next grid plane should extend. This information is obtained with a grid-fitting scheme which monitors the farfield flow structure at each marching station and keeps the outer edge of the grid in the freestream. The grid-fitting scheme and the grid projection techniques are described in more detail later. In the next section, a slightly different three-dimensional formulation of the grid adaptation equations is presented which incorporates a single torsional force term and provides improvements in grid smoothness and algorithm robustness.

7.2 Torsional Dependence, Re-Visited

By performing a force balance along a constant ζ-line on the resulting system of tensional and torsional springs about the point i, j in Fig. 7.1, the following system of equations is obtained,

$$w_i(s_{i+1} - s_i) - w_{i-1}(s_i - s_{i-1}) - K\theta_i = 0 \tag{7.1}$$

where, following the notation of preceding chapters, s_i is the resulting length of the current adaptation line up to the point i, j in Fig. 1 and K is a proportionality constant. The torsion angles, θ_i, can be approximated using the notation of Fig. 7.1 as $\theta_i \approx (s_i - s_i')/|DB|$. The quantities s_i' are defined as the total length of the coordinate line up to the location of the grid reference location.

The term $K\theta_i$ represents a force on the grid point towards the reference point corresponding to s_i'. The formulation of Chapter 5 consist of multiple torsional terms which can often produce opposing torsional forces and cause inconsistent grid point movement. At first glance, comparing Eq. (7.1) with the formulation of Eq. (5.1) in Chapter 5, it would appear that only the cross-stream influence is present in Eq. (7.1). However, the single proportionality constant K of the present formulation now contains grid information from both streamwise and

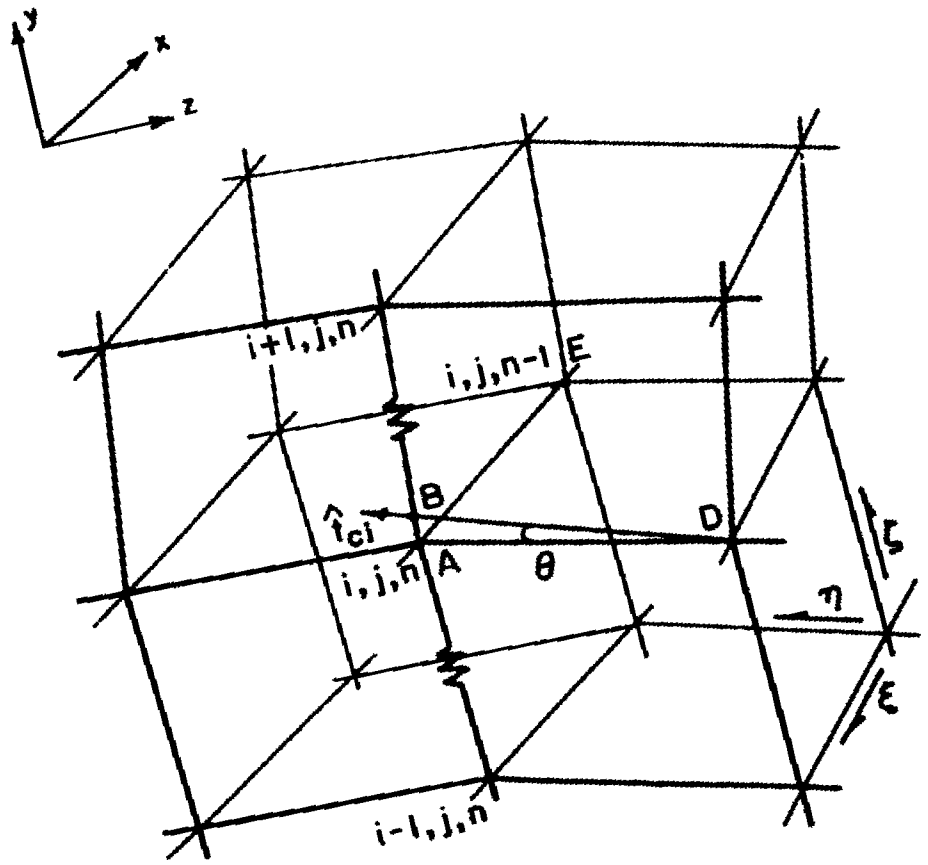


Figure 7.1: Adaptive mesh geometry.

cross-flow directions and its calculation is described in the next section. The use of a single torsion term in Eq. (7.1) is found to produce a smoother adaptive grid, less sensitive to the difference in grid scales in the streamwise and cross-flow directions.

Substitution of the expression for θ_i into Eq. (7.1), rearranging and defining $\tau_i = K/|DB|$ the following linear system is obtained.

$$w_{i-1}s_{i-1} - (w_{i-1} + w_i + \tau_i)s_i + w_i s_{i+1} = -\tau_i s_i' \tag{7.2}$$

7.3 Torsional Force Term

Equation (7.2) differs from Eq. (5.2) in that a single torsional term appears. This approach has distinct advantages over the multiple torsional approach of Chapter 5.

Equation (7.2) indicates that as the term τ_i becomes large, the equilibrium position of the ith grid point approaches the position corresponding to the location s_i'. This location represents the equilibrium position of the individual torsional springs projecting from neighboring grid lines.

Consequently, a value of s_i' much different from $\bar{s}_i$, the positions of the grid points before grid adaptation, can result in excessive movement of grid points, even without any input from the error measure. The initial grid point position, $\bar{s}_i$, before the grid adaptation step, corresponds to the grid point position that results from the downstream projection of the grid at the $n-1$ marching station to the nth marching station. The grid surface at the nth or current marching station combined with that at the $n-1$ marching station make up two sides of a computational slab. As discussed previously, before adaptation and during the preliminary solution process, the downstream side of this computational slab is simply a proportioned duplicate of the upstream side. After adaptation, the two sides of the computational slab should be sufficiently aligned with one another to

permit an improved solution during the re-marching step of the solution-adaptive procedure. If excessive skewing occurs, the solution process becomes unstable. What is desired is that, for sufficiently large values of τ_i, the grid continually retains the smoothness necessary for obtaining an improved solution on the adapted grid.

In order to achieve precise and bounded grid point movement, the values of s_i' must remain close to the preliminary grid point positions, $\bar{s}_i$, and, at the same time, the values of τ_i of Eq. (7.2) must be chosen so as to hold the grid close to these locations. Values of τ_i too large can smother the effect of the weighting functions, causing the grid to move independent of the solution while values too small permits excessive movement.

To make all the terms of Eq. (7.2) approximately equal in magnitude at each point along the adaptation line, τ_i is defined in the following manner,

$$\tau_i = \lambda(w_{i-1} + w_i)\bar{\alpha}_i \tag{7.3}$$

where λ is a user defined stiffness parameter and $\bar{\alpha}_i = \alpha_i/\alpha_{min}$ is the aspect ratio of the grid cell normalized with respect to the minimum cell aspect ratio. Each aspect ratio is computed using the notation of Fig. 7.1 as follows,

$$\alpha_i = \frac{s_{i+1,j,n} - s_{i-1,j,n}}{|DB|}$$

The use of Eq. (7.3) in Eq. (7.2) ensures that all terms of Eq. (7.2) are of the same magnitude for all points along the line. With this formulation, equal consideration is given to both the weighting functions and the surrounding grid geometry in deciding how the grid points should move.

Also, another desirable feature of incorporating only one torsion term in Eq. (7.2) which utilizes a weighted averaging for determining the grid reference positions, s_i', is that a wider range of values for λ exist for which the second solution step of the solution-adaptive procedure remains stable.

The preliminary grid point positions, $\bar{s}_i$, obtained by projecting the upstream grid plane to the current location (point A of Fig. 1) do not necessarily satisfy the requirements for grid orthogonality and straightness in the cross-flow direction. For satisfying orthogonality and straightness requirements in the cross-flow direction, grid torsional reference positions, $s'_{c\,i}$, can be defined, as described later. The net torsional reference positions, s'_i, of Eq. (7.2) are the positions the grid points should take based on considerations of orthogonality and straightness both in the streamwise direction (specified by $\bar{s}_i$) and in the cross-flow direction (specified by $s'_{c\,i}$). Thus, the net torsional reference positions, s'_i are defined as

$$s'_i = \bar{s}_i + \epsilon(s'_{c\,i} - \bar{s}_i) \tag{7.4}$$

where ϵ is a proportionality constant. As mentioned in the preceding section, the τ_i values can be so chosen that the solution to Eq. (7.2) yields s_i values that are in the neighborhood of the corresponding s'_i values. A value of zero assigned to ϵ neglects cross-flow torsional dependence while a value of unity permits the grid to move without regard to the streamwise grid surface. Obviously, either of these extremes would quickly produce unacceptable results. Small values of ϵ produce an acceptable degree of grid movement with respect to the grid plane upstream of the current marching station, resulting in sufficient alignment between the two grid planes. However, values too small can cause grid irregularities in the cross-flow direction. A value of $\epsilon = .3$ was used for all computations in the section to follow.

The quantities $s'_{c\,i}$ represent the positions the ith grid point must take in order that the grid acquire the desired amounts of straightness and orthogonality in the cross-flow direction. The position $s'_{c\,i}$ is computed in exactly the same manner as presented in Chapter 5 and is defined as the intersection point of the cross-flow torsion vector $\hat{\mathbf{t}}_{c\,i}$ and the current adaptation line (point B of Fig. 1).

The cross-flow torsion vector, $\hat{\mathbf{t}}_{ci}$, is also computed in the same manner as in Chapter 5 by combining user specified contributions of grid orthogonality and straightness in the cross-flow direction. Equation (5.4) is repeated here for convenience

$$\hat{\mathbf{t}}_{ci} = C_c \hat{\mathbf{s}}_{ci} + (1 - C_c)\hat{\mathbf{n}}_{ci}$$

where C_c can range from zero to unity. A value close to zero enforces more orthogonality while values closer to unity provide more grid straightness. The straightness vector, $\hat{\mathbf{s}}_{ci}$, is defined as a vector from the point $(i, j-2, n)$ to the point $(i, j-1, n)$ of Fig. 7.1. The orthogonality vector, $\hat{\mathbf{n}}_{ci}$, is defined as an average of vectors normal to the current grid line and the neighboring line in the cross-flow plane.

To insure that s'_{ci} is monotonically increasing and to provide further grid point motion control, the point B is limited to $\frac{1}{2}\Delta s_k$, where Δs_k is either the distance from the point $(i+1, j, n)$ to the point (i, j, n) if the torsional vector crosses above the point A as in Fig. 1 and is the distance from the point (i, j, n) to the point $(i-1, j, n)$ if $\hat{\mathbf{t}}_{ci}$ crosses below point A. This procedure was also presented in Chapter 5.

7.4 Three-Dimensional Grid-Fitting Procedure

This section outlines the procedures by which the total height of the grid is estimated at each marching station and the steps used to project the grid plane at the current marching station to the next marching station. This feature of the present procedure, while presented in earlier sections for two-dimensional flows, required major extensions for the three-dimensional situations of interest in this chapter. The solution-adaptive grid-fitting algorithm begins with an initial grid plane of data at the current marching station. This grid plane must be projected downstream to form a computational slab on which numerical integration can

be performed. The first step in this projection process is to determine how far out into the freestream the next grid plane must extend. Each η-coordinate line at the current marching station is searched to find the innermost position of freestream conditions and is clipped (or extrapolated to) a prescribed distance above this location (This location is illustrated by the solid symbols in Fig. 7.2. The resulting new η-ray lengths are first smoothed to minimize inconsistencies in the position of the outer grid line and then, using a few of these η-line lengths (denoted by the cross symbols of Fig. 7.2) as control points, a spline is fitted to form a temporary new outer grid line at the current marching station. This new grid line is then projected forward to the next marching station using a conical projection from a point far upstream.

At this point in the grid projection process, we now have, in addition to the plane of data at the current marching station and the surface grid, an outer grid line which serves as a farfield grid boundary at the next marching station (See Fig. 7.2). The next task is to project the remainder of the grid plane at the current marching station (upstream side of the computational slab) to the next marching station (downstream side of the computational slab) in such a manner so as to ensure that the two sides of the resulting grid slab are uniformly aligned. Figure 7.3 illustrates a typical grid plane at the current marching station extending outward from the current ξ-section of the surface grid. Solid circles are used to represent grid points in the current marching station and open circles denote grid points (projected points) in the next marching station. To construct the grid plane at the next marching station, each kth point on the ξ_n-line of the surface grid is extended a distance far upstream. For example, the point $\xi_n, \eta_k, 0$ of the surface grid is extended to the point p_k of Fig. 7.3, which is defined here as the kth projection point, by extrapolating upstream the straight line segment joining the surface points $\xi_{n+1}, \eta_k, 0$ and $\xi_n, \eta_k, 0$. In this manner, each of the

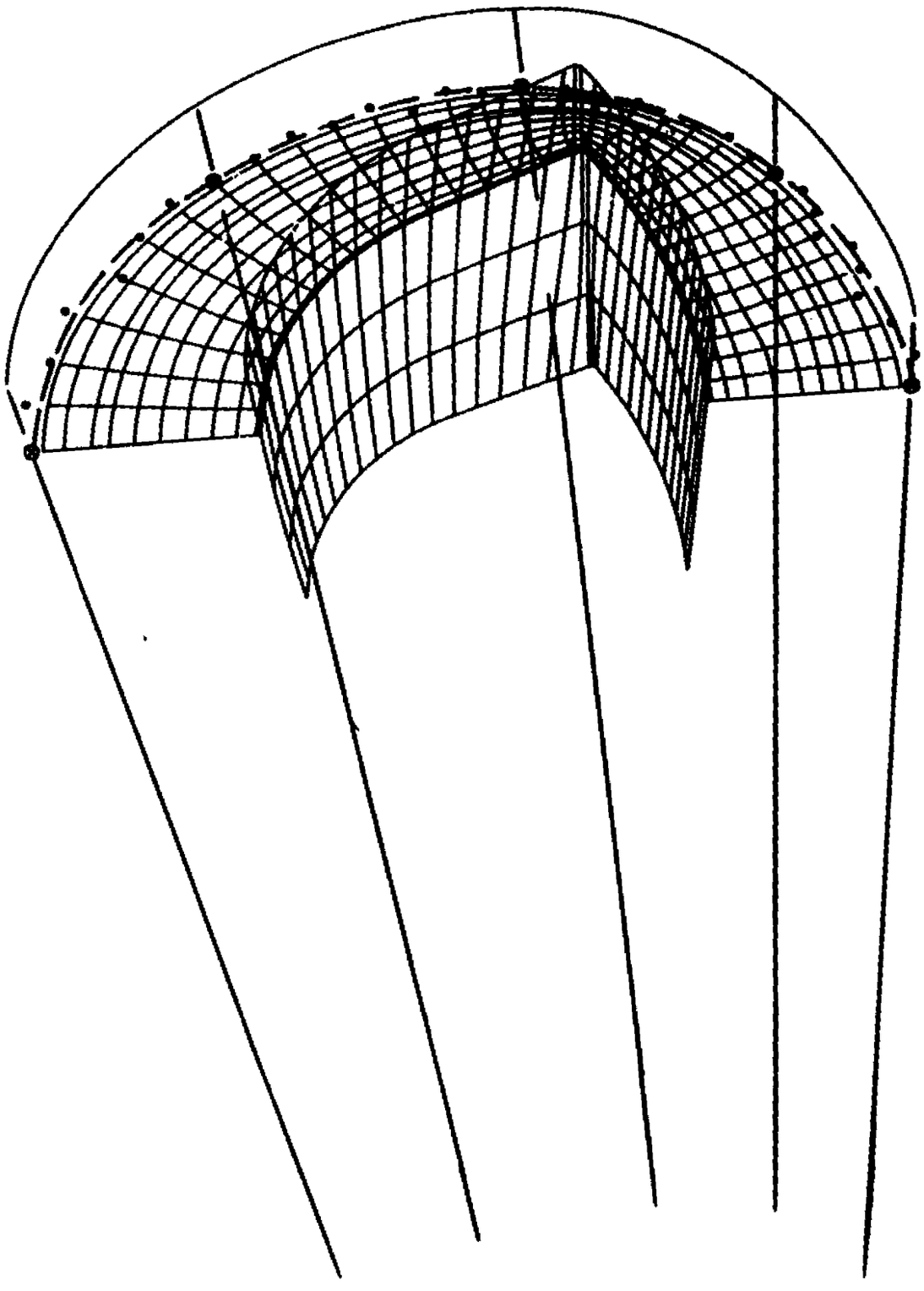


Figure 7.2: Schematic of grid-fitting procedure.

surface grid points in the current marching station at ξ_n are projected backwards to construct corresponding projection points. All of these projection points lie in a common ξ =constant plane far upstream. The kth projection point, p_k, is then used as a reference from which each grid point along the kth η-line in the current marching station is projected linearly into the next marching station. This is done by extending a ray from the kth projection point of Fig. 7.3, through each of the ξ_n, η_k, ζ_l grid points along the η_k-line in the current marching station to a distance $\Delta\xi$ downstream.

If the plane of projection points is sufficiently far upstream such that $\Delta\xi/(\xi_n - \xi_p) \ll 1$, where ξ_p is the ξ-position of the plane containing the projection points, then the resulting η-line projected into the downstream marching plane will have the same shape as the corresponding η-line in the current marching plane. Once all the η-lines have been projected downstream in this fashion, a stretching procedure is performed to connect the outer edge of the grid at the new station to the grid-fitted outer line obtained as described previously. Recall that the grid-fitted outer line is the new position of the farfield grid boundary as determined by the grid-fitting procedure, each point of which was projected downstream from a *common* point far upstream.

In addition to providing an estimated height above the body for projecting the grid downstream, the grid-fitting scheme has been found to assist in providing improvements in the performance of the adaptive grid algorithm. For complex three-dimensional external flows, grid skewing problems can arise when adapting the grid using the adaptive grid techniques outlined in the previous section if the shape of the grid differs too severely from that of the evolving bow shock. If a large number of grid points are initially in the freestream, outside the bow shock, then grid points will cluster in the vicinity of the bow shock. Any points outside the shock are in a region of constant flow-field properties (i.e. freestream

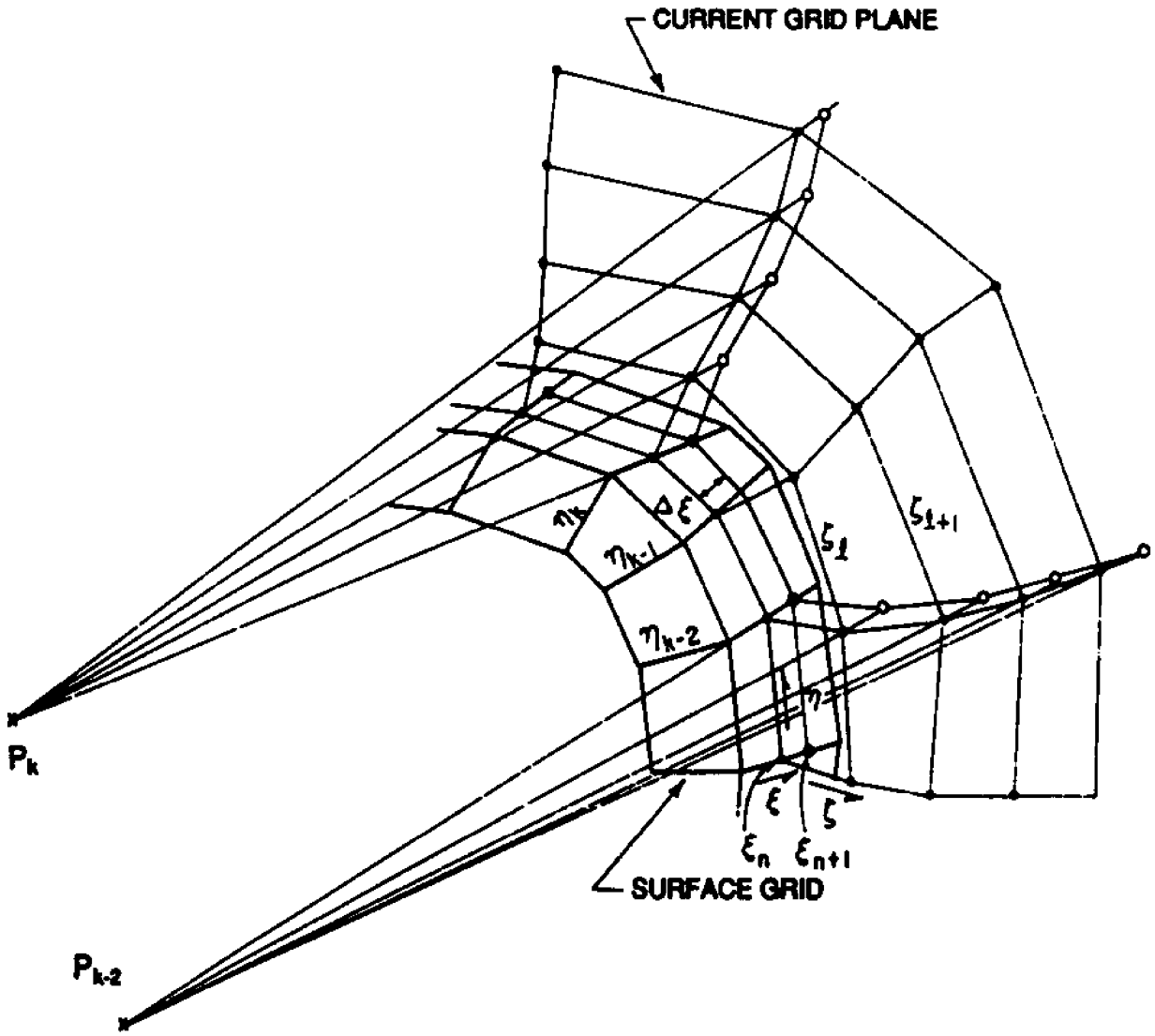


Figure 7.3: Downstream projection of adapted grid plane.

conditions). As the solution proceeds downstream and the adaptation procedure draws grid points into the steep gradient of the bow shock, the outer grid lines tend to kink. The use of the grid-fitting scheme keeps the shape of the grid accurately aligned with the outer shock, producing a degree of resolution that can only be rivaled with the use of a shock-fitting algorithm. This advantage brought about by the use of the grid-fitting algorithm will be made more apparent with the use of illustrations in the results section to follow.

7.5 Ames All-Body Vehicle

The three-dimensional solution-adaptive PNS algorithm is tested for the hypersonic flow over the all-body configuration with elliptical cross-sections shown in Fig. 7.4. The flow-field conditions were chosen to match those of the experimental and computational study conducted by Lockman, Lawrence and Cleary, (1988) These conditions are:

$$M_\infty = 7.4 \qquad \tilde{T}_\infty = 61.5K \qquad \gamma = 1.4$$

$$Re_L = 15 \times 10^6 \qquad \tilde{T}_w = 311.1K \qquad Pr = 0.72$$

where $L = .9144m$ is the total length of the model. A perfect gas is assumed for the present calculations. The flow is turbulent and the eddy viscosity is modeled using the algebraic turbulence model of Baldwin and Lomax, (1978) with transition to fully turbulent flow assumed at $x/L = 0.05$.

For each angle of incidence, a conical stepback procedure was performed on an initial grid plane at $x/L = .05$ obtained by a hyperbolic grid generation routine. Once a converged conical solution was obtained, the grid-fitting algorithm was used to fit the outer grid line to the shock. The conical stepback procedure was then allowed to continue with the use of solution-adaptive gridding for about 100 iterations. Afterwards, grid adaptation was terminated and the conical stepback procedure was allowed to continue on the fixed (adapted) starting plane until

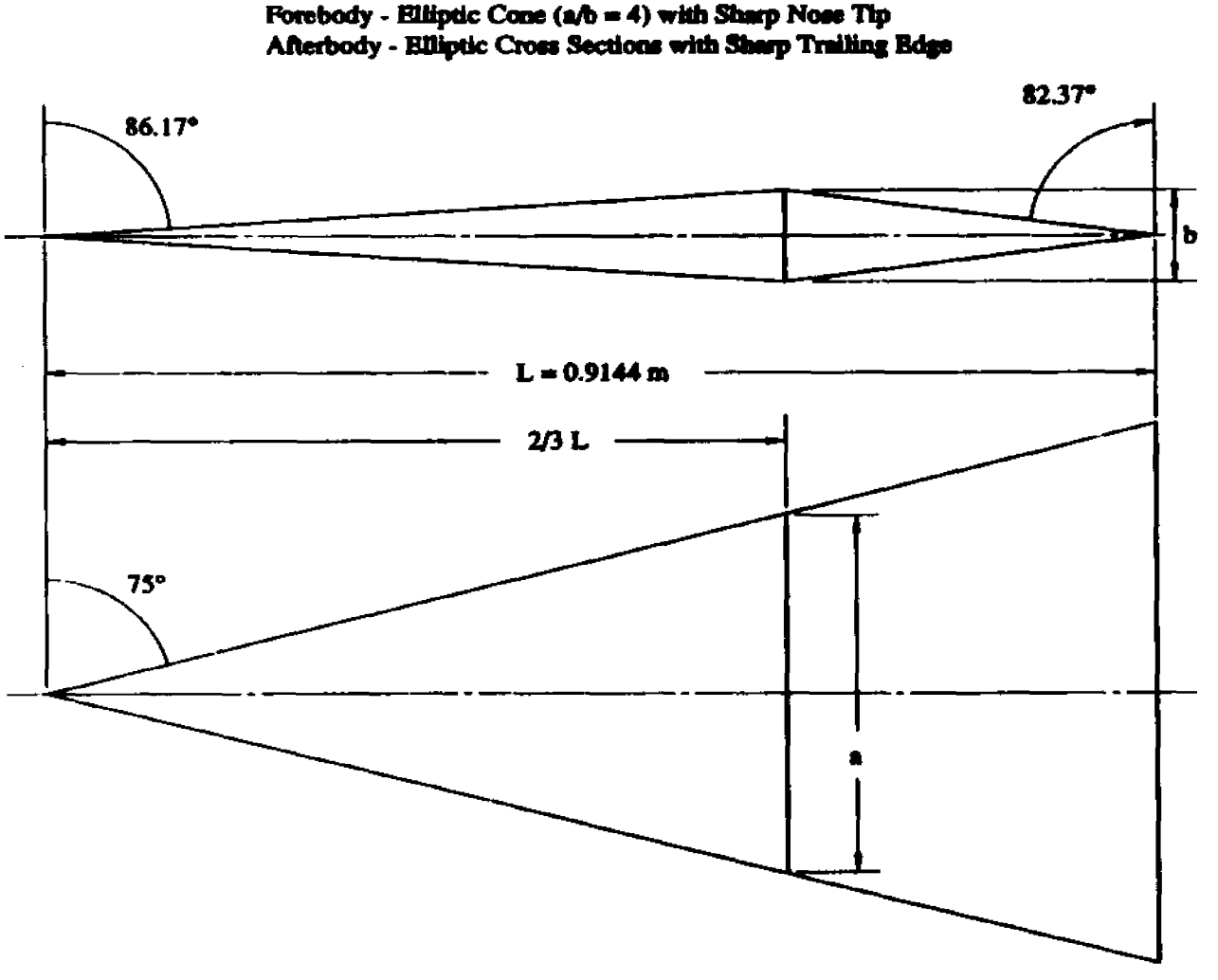


Figure 7.4: Allbody hypersonic vehicle geometry.

a converged adapted plane of data was obtained. For both angle of incidence cases each cross-flow grid plane consist of 151 grid points in the circumferential direction and 90 grid points in the radial direction.

All adaptation parameters used for the present studies were given identical values for both angle of incidence cases. Also, for each angle of incidence, Mach number, pressure and density were used as the adaptation variables. The maximum and minimum allowable grid spacings where given the following values: $(\Delta s_{MAX})_\zeta/\Delta s_{ave} = 2.5$, $(\Delta s_{MIN})_\zeta/\Delta s_{ave} = .5$, $(\Delta s_{MAX})_\eta/\Delta s_{ave} = 2.0$ and $(\Delta s_{MIN})_\eta/\Delta s_{ave} = .5$, where Δs_{ave} is the average grid spacing along the adaptation line, C_ζ and C_η are the grid straightness to orthogonality parameters (Eq. 7) used for the radial and circumferential adaptations and where given values of 0.7 and 0.4, respectively. The torsional parameters, λ_ζ and λ_η of Eq. (6), were given values of 2×10^{-3} and 25, respectively. Quite frequently, optimum values of these parameters for adaptation in different cross-flow directions can vary. The η- and ζ-subscripts indicate parameters used for grid adaptation sweeps in the respective directions.

After a suitable starting solution was obtained using the stepback procedure, the solution-adaptive, grid-fitting and marching procedure was then employed with a marching stepsize of 10^{-3} to a x/L-location of 0.8 for each angle of incidence case. Circumferential adaptations were performed by sweeping from the body surface outward while the direction of the radial adaptation sweeps alternated. Sweeping one marching station from windward to leeward side and the next station from leeward to windward. This alternation in the direction of the radial adaptations improves grid smoothness, allowing grid orthogonality and straightness information to propagate from both sides of the grid.

Figure 7.5 consists of a sectional view of the resulting solution-adapted and grid-fitted computational grid for the 0° angle of incidence case. This figure

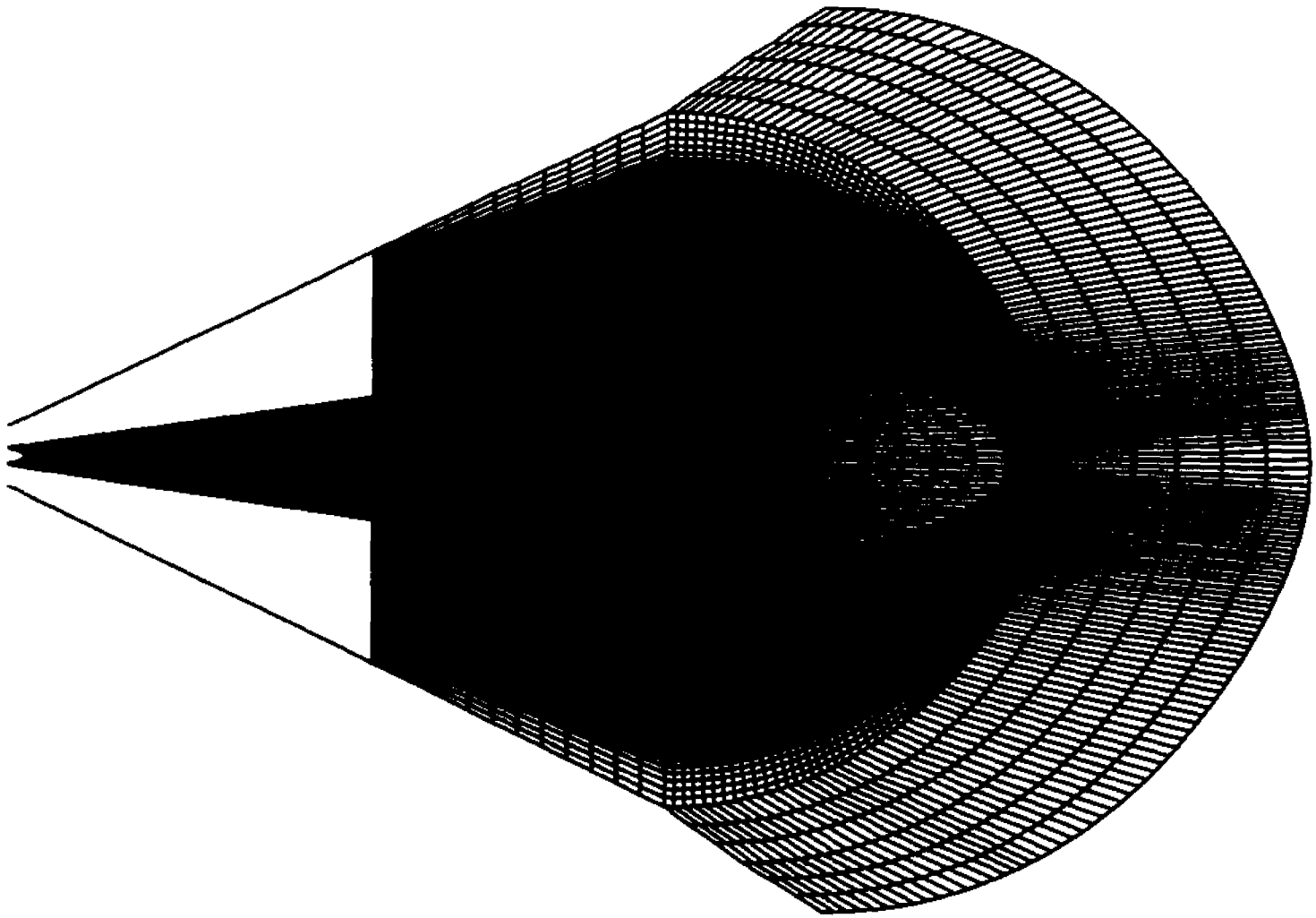


Figure 7.5: Sectional view of solution-adapted and grid-fitted computational grid.
($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, 151x90 crossflow grid)

illustrates the algorithm's ability to cluster points in regions of interest and accurately align circumferential grid lines with the cross-flow flow-field structure while also increasing grid point density. The farfield edge of the grid is accurately aligned with the existing bow shock, the position of which is indicated by increased grid clustering. Figure 7.6 illustrates a comparison of cross-flow grid planes at $x/L = .667$ for fixed grid and solution-adaptive cases. This figure further illustrates the grid-fitting algorithm's ability to increase grid point density in virtually all regions of the flow by monitoring the position of the bow shock as the solution proceeds and adjusting the shape of the outer grid line accordingly. For the solution-adapted case, the circumferential grid lines have clustered around the location of the bow shock as well as in the boundary-layer.

The strategy in selecting the adaptation parameters can be best explained with reference to Fig. 7.6. The radial torsion parameter, λ_ζ, was set relatively small (less resistance to grid point movement), and at the same time, the radial grid straightness parameter, C_ζ, was set so as to enforce more grid straightness. At the same time, the circumferential torsion parameter, λ_η, was set to a higher value, with the circumferential grid straightness parameter, C_η, set to enforce more orthogonality. Using this set of parameters, grid points were permitted to move more freely during the radial adaptation sweeps while more grid orthogonality is enforced on the circumferential sweeps. If strict orthogonality were imposed during both directional sweeps, more restriction in point movement occurs, resulting in a less adapted grid.

Figure 7.7 shows Mach number contour comparisons at this same cross-flow location. Significant improvement in the resolution of the bow shock can be seen. Small oscillations in the contours at the maximum spanwise location (near the cross-flow stagnation point) can be seen in the fixed grid half of the figure. These oscillations are not present in the solution-adapted half. Similar improvements in

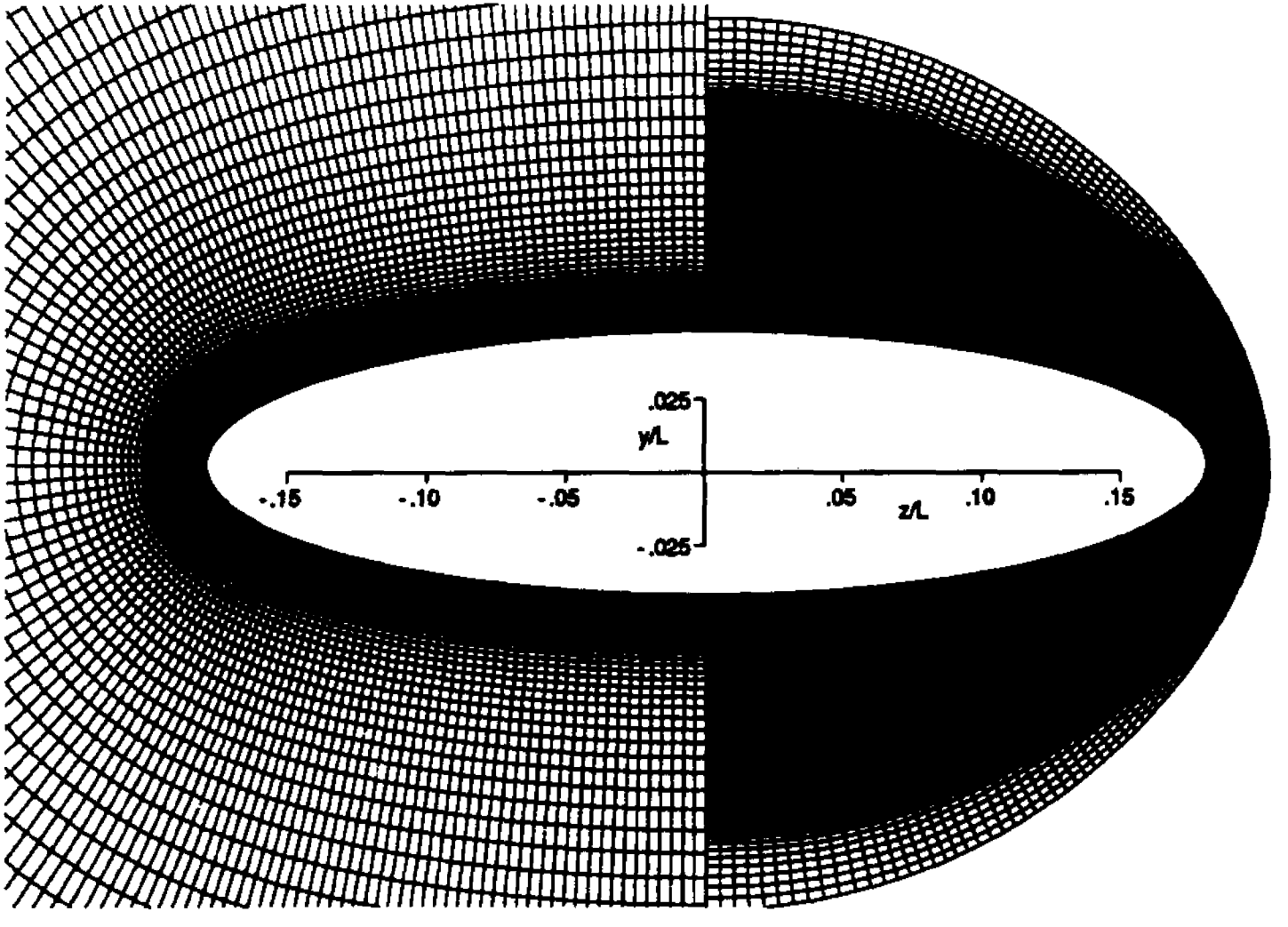


Figure 7.6: Comparison of crossflow computational grid planes; left-half, fixed grid; right-half, solution-adapted and grid-fitted. (α= 0°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667, 151x90 grids)

flow-field resolution are also evident in Fig.7.8 which compares pressure contours at this same x/L location.

Figure 7.9 illustrates a sectional view of the resulting grid for the 5° angle of incidence case on which the first visible cross-flow grid plane is at $x/L = .667$ (the beginning of the afterbody). The extent of the grid has increased on the leeward side due to the increased angle of attack. The grid-fitting algorithm has effectively aligned the grid with the bow shock assisting the solution-adaptive algorithm in producing a smooth computational grid from surface to freestream on all grid surfaces. A small clustering region is observed on the leeward side near the body surface marking the position of the thickening boundary-layer due to the angle of incidence.

Figure 7.10 compares the cross-flow grid plane at $x/L = .667$ for the solution-adaptive and fixed grid cases. This figure further illustrates the present algorithm's ability to effectively concentrate the majority of the grid in the region of solution variation. Grid orthogonality was achieved during the circumferential adaptation sweep by careful selection of adaptation parameters (as discussed above for the 0° angle of incidence case).

Figure 7.11 compares Mach number contours at this same cross-flow station. A significant increase in bow shock resolution is observed in the solution-adapted case. Improved resolution is also noted in the solution-adaptive results near the cross-flow stagnation point which has moved slightly to the windward side due to the higher angle of incidence. The thickened boundary-layer on the leeward side is also more clearly observed. A comparison of the computed pressure field at this same x/L location is illustrated in Fig.7.12.

Figure 7.13 compares the solution-adapted and grid-fitted cross-flow grid plane of Fig. 7.10 with a grid plane obtained without the the use of the grid-fitting algorithm. For the left half of the figure (without grid-fitting), the clustering at

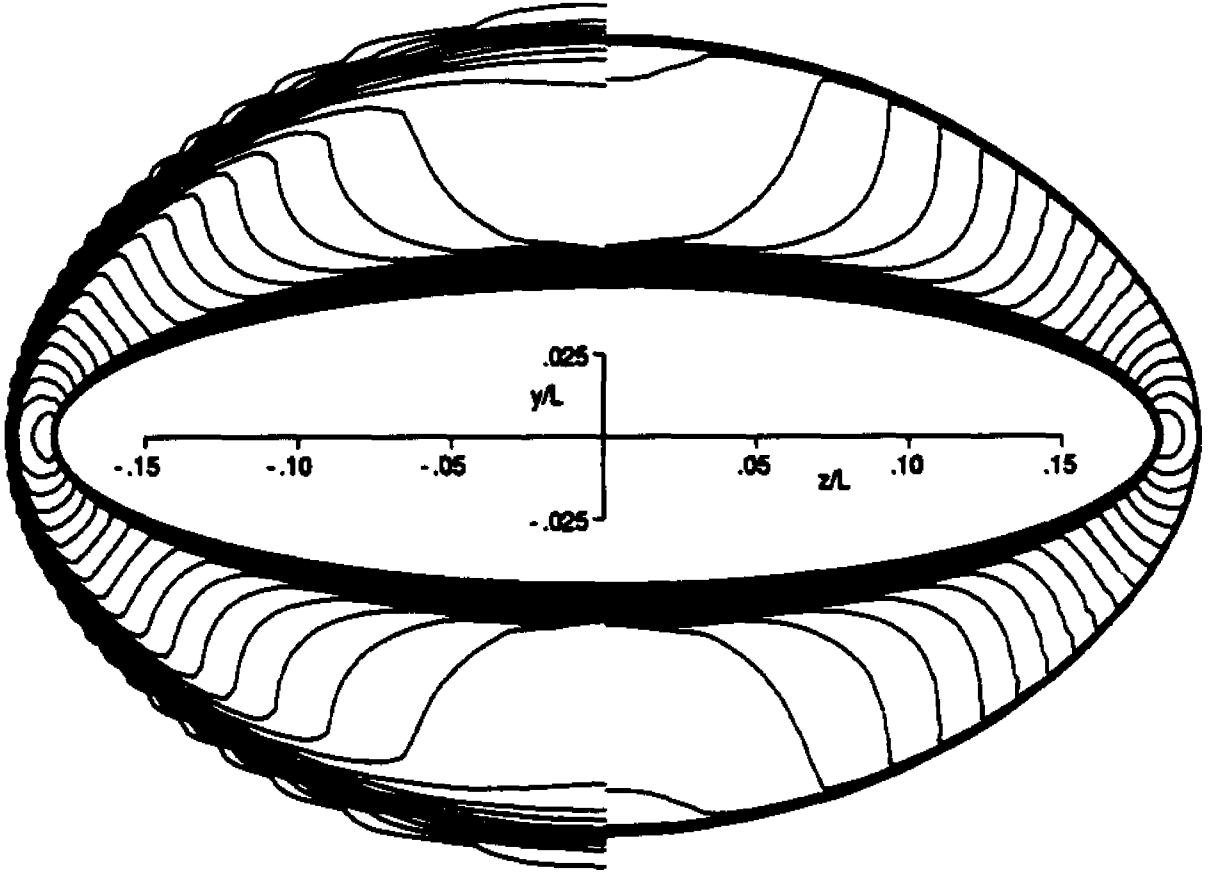


Figure 7.7: Comparison of computed Mach number contours; left-half, fixed grid; right-half, solution-adapted and grid-fitted. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)

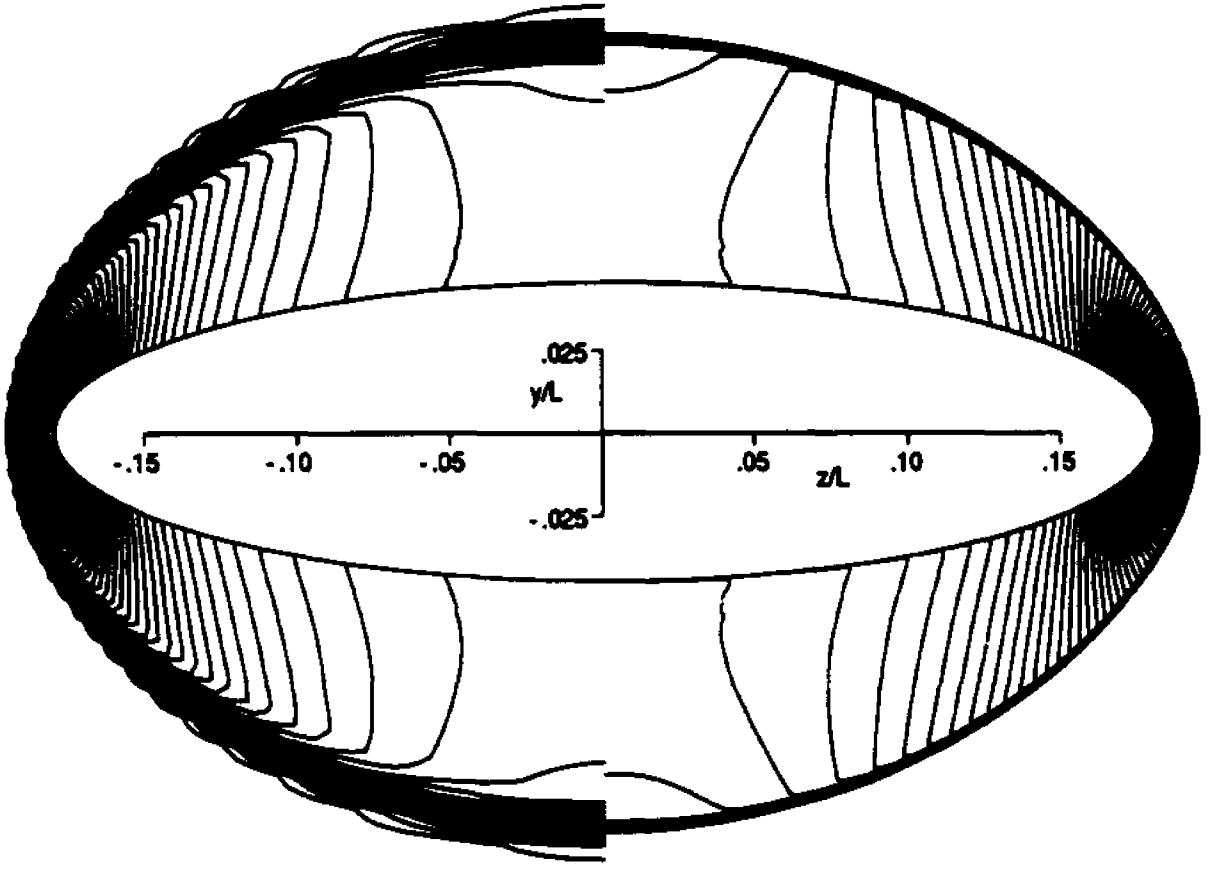


Figure 7.8: Comparison of computed pressure contours; left-half, fixed grid; right-half, solution-adapted and grid-fitted. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$, $x/L = .667$)

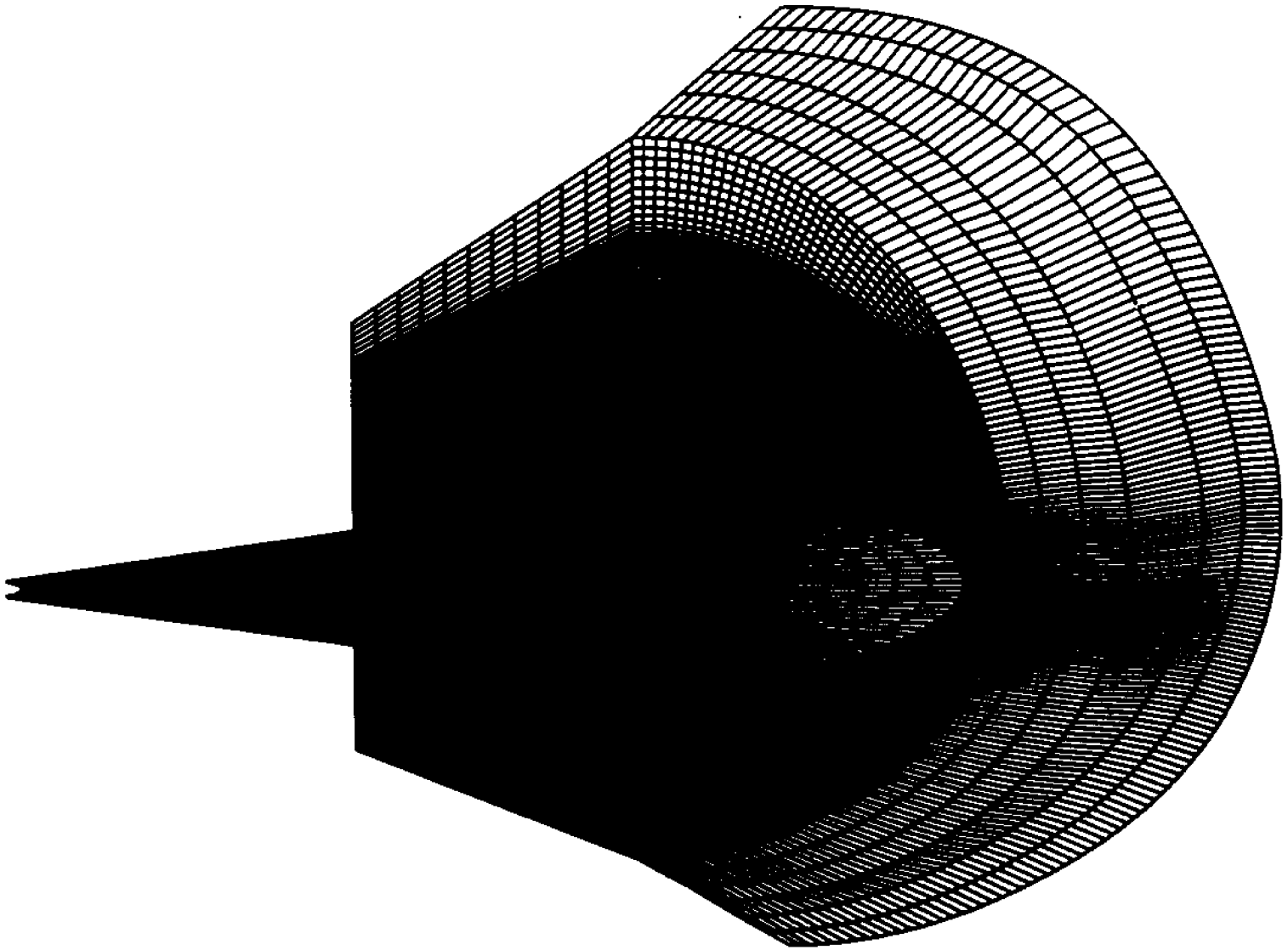


Figure 7.9: Sectional view of solution-adapted and grid-fitted computational grid. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, 151x90 crossflow grid)

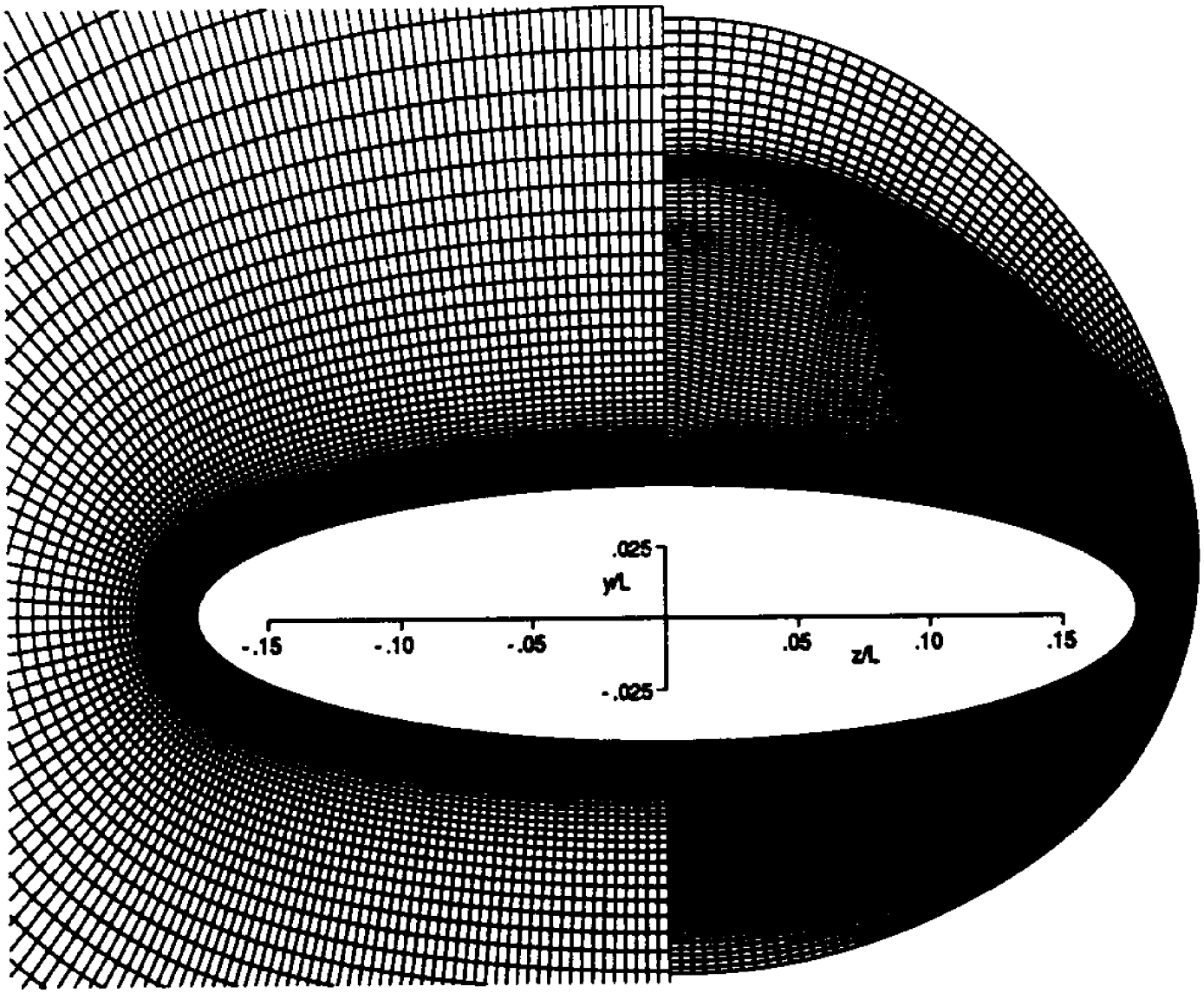


Figure 7.10: Comparison of crossflow computational grid planes; left-half, fixed grid; right-half, solution-adapted and grid-fitted. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667, 151x90 grids)

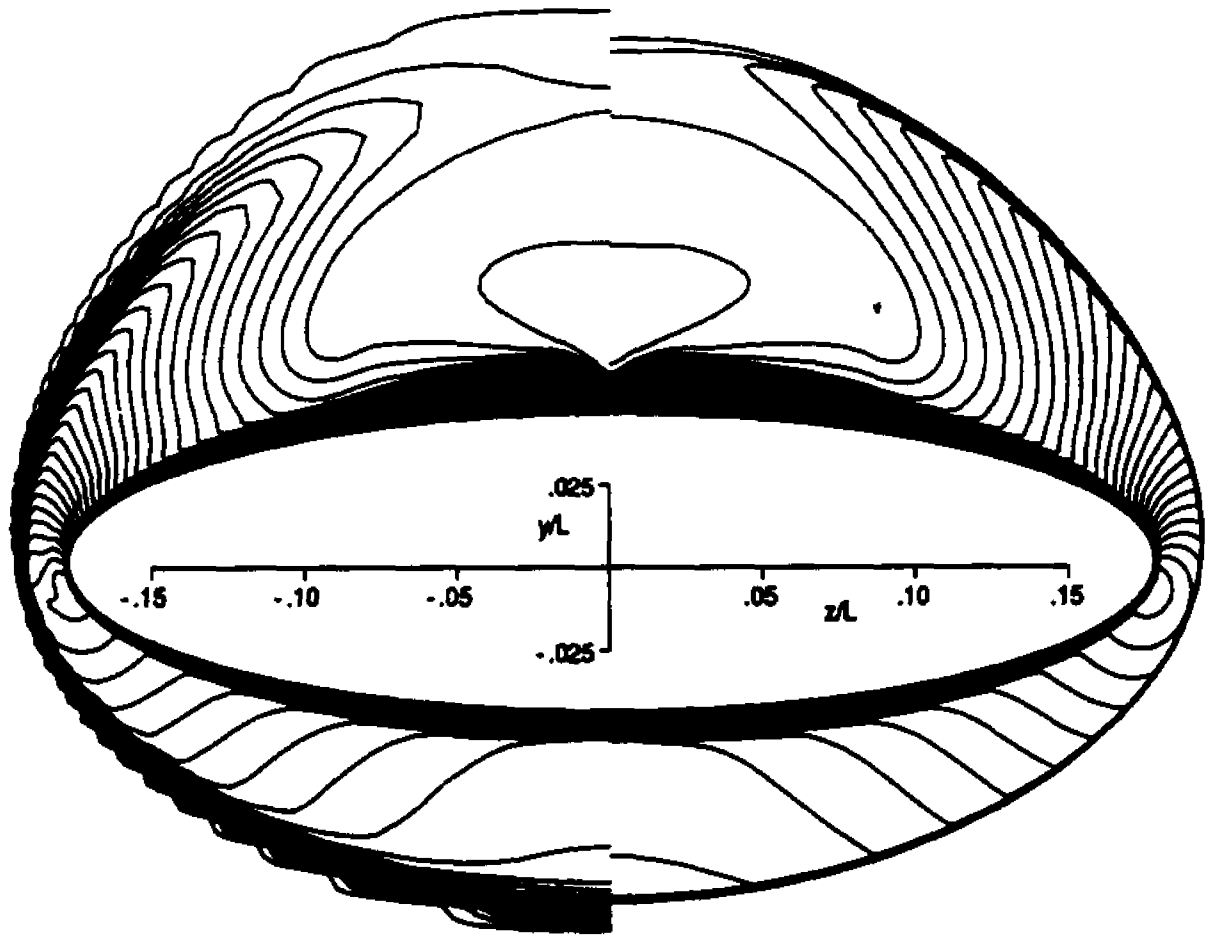


Figure 7.11: Comparison of computed Mach number contours; left-half, fixed grid; right-half, solution-adapted and grid-fitted. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667)

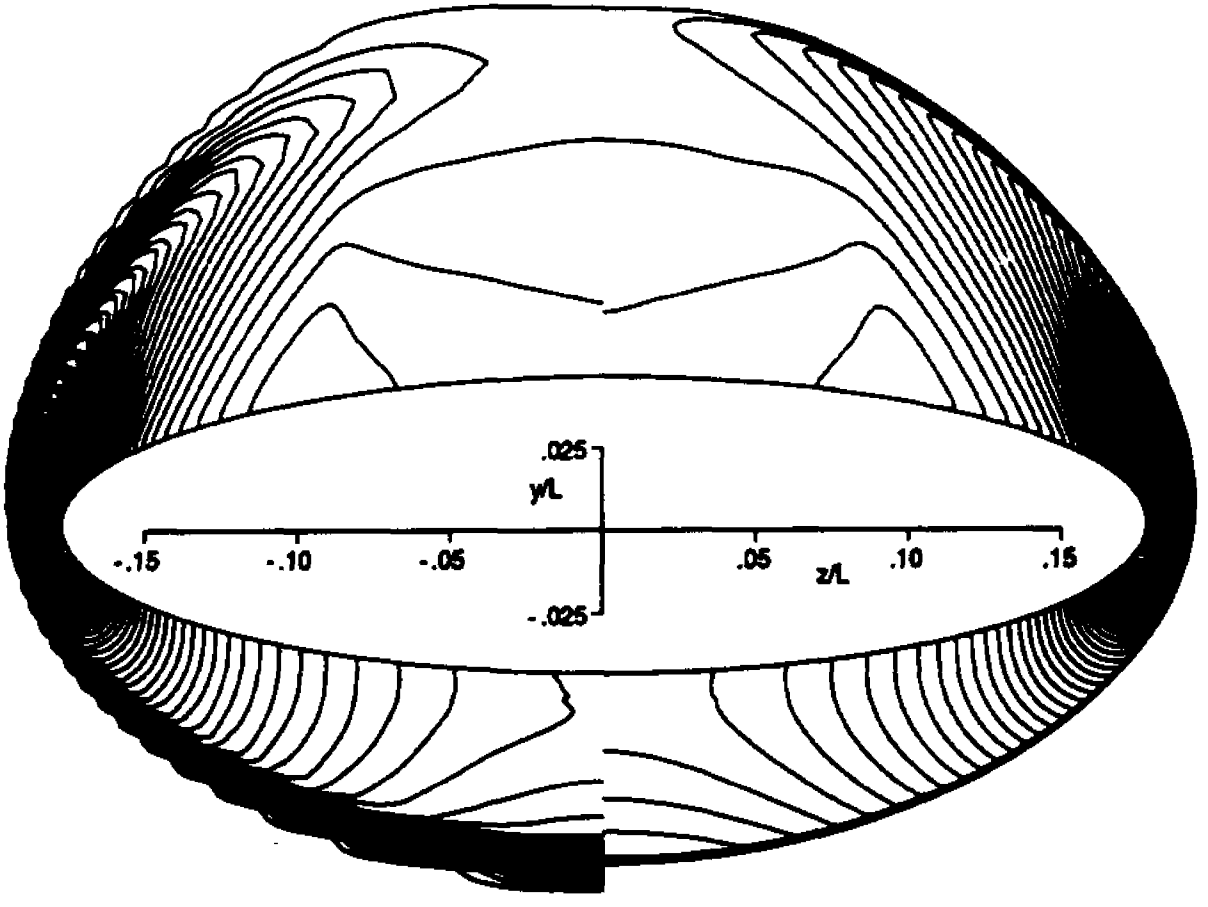


Figure 7.12: Comparison of computed pressure contours; left-half, fixed grid; right-half, solution-adapted and grid-fitted. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667)

the bow shock has caused severe grid skewing. If the outer boundary of the grid is not grid-fitted, then the radial extent of the freestream region compared with the radial extent of the region of interest can vary significantly in the circumferential direction. This is clearly seen in the left-half of Fig. 7.13 where, along the major axis, the extent of the freestream region is much larger than at other circumferential locations. Since the number of points in the freestream do not correspondingly increase with the freestream extent, the radial mesh spacing in the freestream along a radial outward line will be different at different circumferential locations. This causes the mesh in the freestream to distort, particularly when the radial extent of the freestream changes appreciably in the circumferential direction. Clearly, this is unacceptable, as seen in the left-half of Fig. 7.13.

The grid-fitting algorithm eliminates these problems by producing a cross-flow grid plane whose shape is similar to that of the overall flow disturbance, thereby minimizing the number of points in the freestream and maintaining approximately an equal number of them all the way around the outside of the grid.

There is another significant difference in how these two grid solutions were obtained. For the solution-adapted and grid-fitted case (right-half of Fig. 7.13), only one torsional force term was used in the grid adaptation equation (Eq. 7.2). For the left-half of Fig. 7.13, the formulation of Chapter 5 was used which includes two torsion force terms. These two terms, as discussed previously, can often oppose one another, producing grid skewness, the effects of which can be seen by comparing the differences in grid smoothness inside the bow shock (right and left halves of Fig. 7.13).

Grid skewness caused by the adaptation algorithm can adversely effect the quality of the solution. This is illustrated in Fig. 7.14 which compares computational grids and Mach contours at $x/L = .667$ near the cross-flow stagnation point. Grid lines bulging outward from the bow shock region in Fig. 7.14*b* result

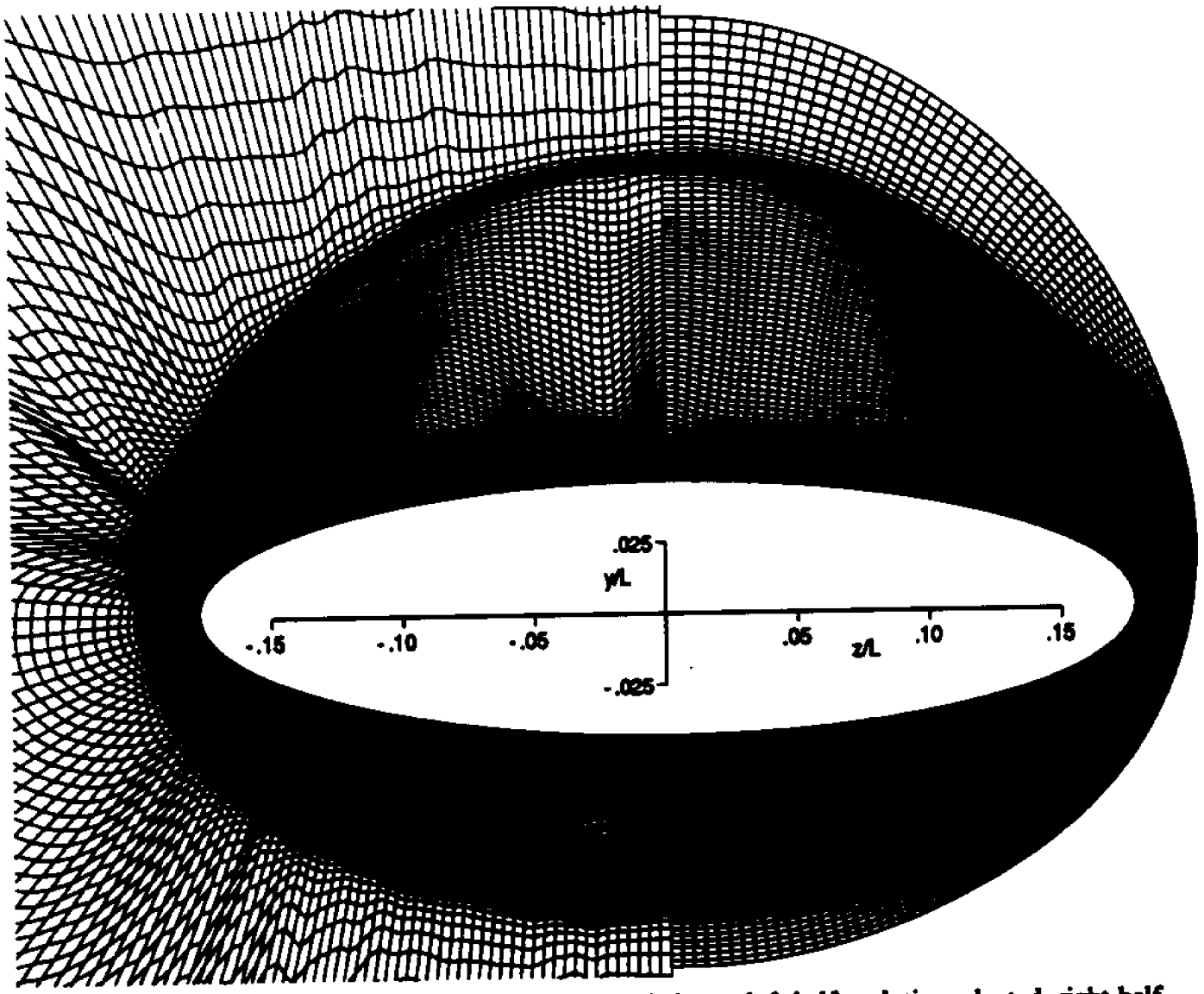


Figure 7.13: Comparison of crossflow computational grid planes; left-half, solution-adapted; right-half, solution-adapted and grid-fitted. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667, 151x90 grids)

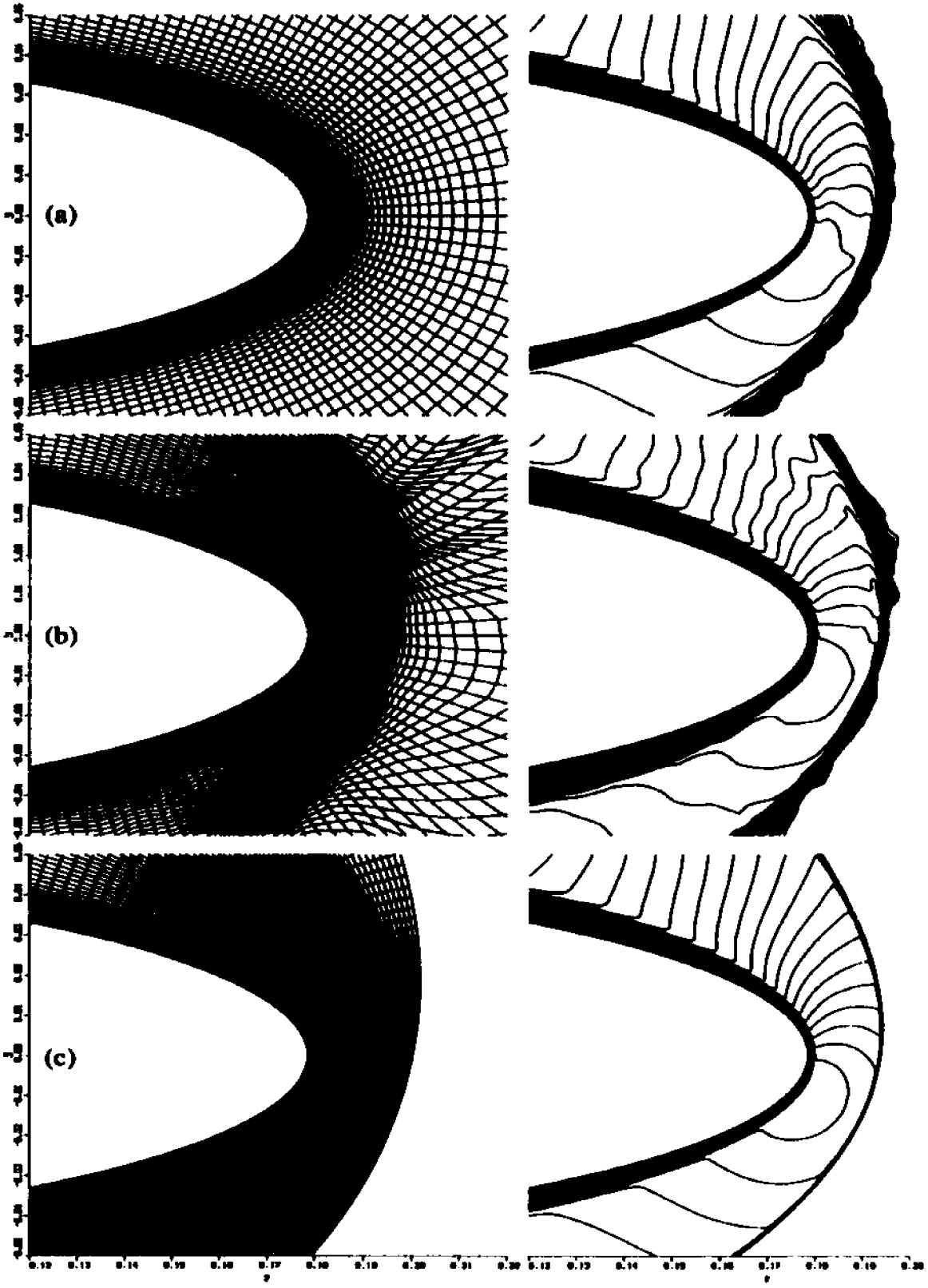


Figure 7.14: Comparison of grids and Mach number contours near crossflow stagnation point. (a) fixed grid, (b) adapted, w/o grid-fitting, (c) adapted, grid-fitted. (α= 5°, M_∞= 7.4, $Re_{\infty,L}$= 15x10^6, x/L= .667, 151x90 grids)

in a grid distortion and inaccuracies in the predicted resolution and smoothness of the shock. In addition, the solution inside the shock also deteriorates. For the solution-adapted and grid-fitted case shown in Fig. 7.14*c*, flow features are sharp and well resolved, illustrating the substantial improvements in accuracy over the solutions shown in Figs. 7.14*a* and 7.14*b*. The increased grid smoothness inside the bow shock of the solution-adapted, grid-fitted case, compared with that in the grid of Fig. 7.14*b*, is attributed to the re-formulation of the adaptation equations to include a single torsion term (as discussed previously).

In Fig. 7.15 predicted pitot pressure profiles obtained with a fixed grid and the solution-adapted, grid-fitting algorithm are compared with experimental data at various x/L-locations for the 0° angle of incidence case. Improvements are seen at each location in the resolution of the bow shock. The solution-adapted and grid-fitted algorithm predicts a shock thickness which is approximately 15 to 20 percent of the thickness of the shock predicted using a fixed grid. Slight improvements can also be detected in the expansion region produced by the afterbody. The experimental data is taken from the study of Lockman et al.[14]

In Fig. 7.16 predicted pitot pressure profiles obtained with a fixed grid and the solution-adapted, grid-fitting algorithm are compared with experimental data at $x/L = .8$ for the various angle of incidence cases studied. Again, significant improvements are detected near the bow shock on the windward side for both angles of incidence. Computed shock thickness for the solution-adapted case on the windward side at 5° angle of incidence is only 10 percent of the shock thickness predicted using a fixed grid. Closer agreement with experiment is also observed for the solution-adapted case compared with the fixed grid case in the expansion region on both the windward and leeward sides of the 5° angle of incidence case.

Figure 7.17 compares surface pressure distributions along the symmetry plane for both angles of incidence. The results using both the solution-adaptive scheme

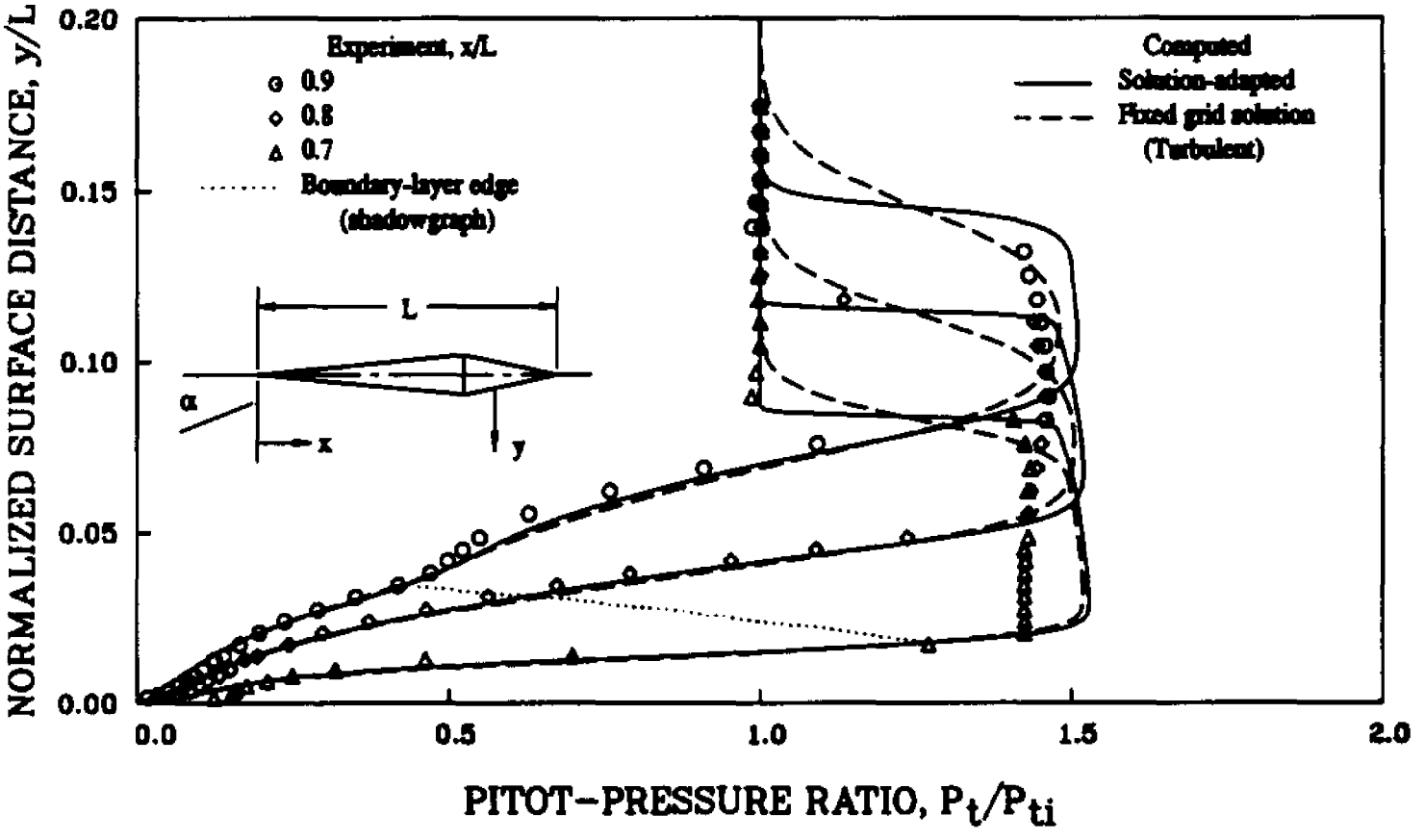


Figure 7.15: Comparison of pitot pressure profiles. ($\alpha = 0°$, $M_\infty = 7.4$, $Re_{\infty,L} = 15 \times 10^6$)

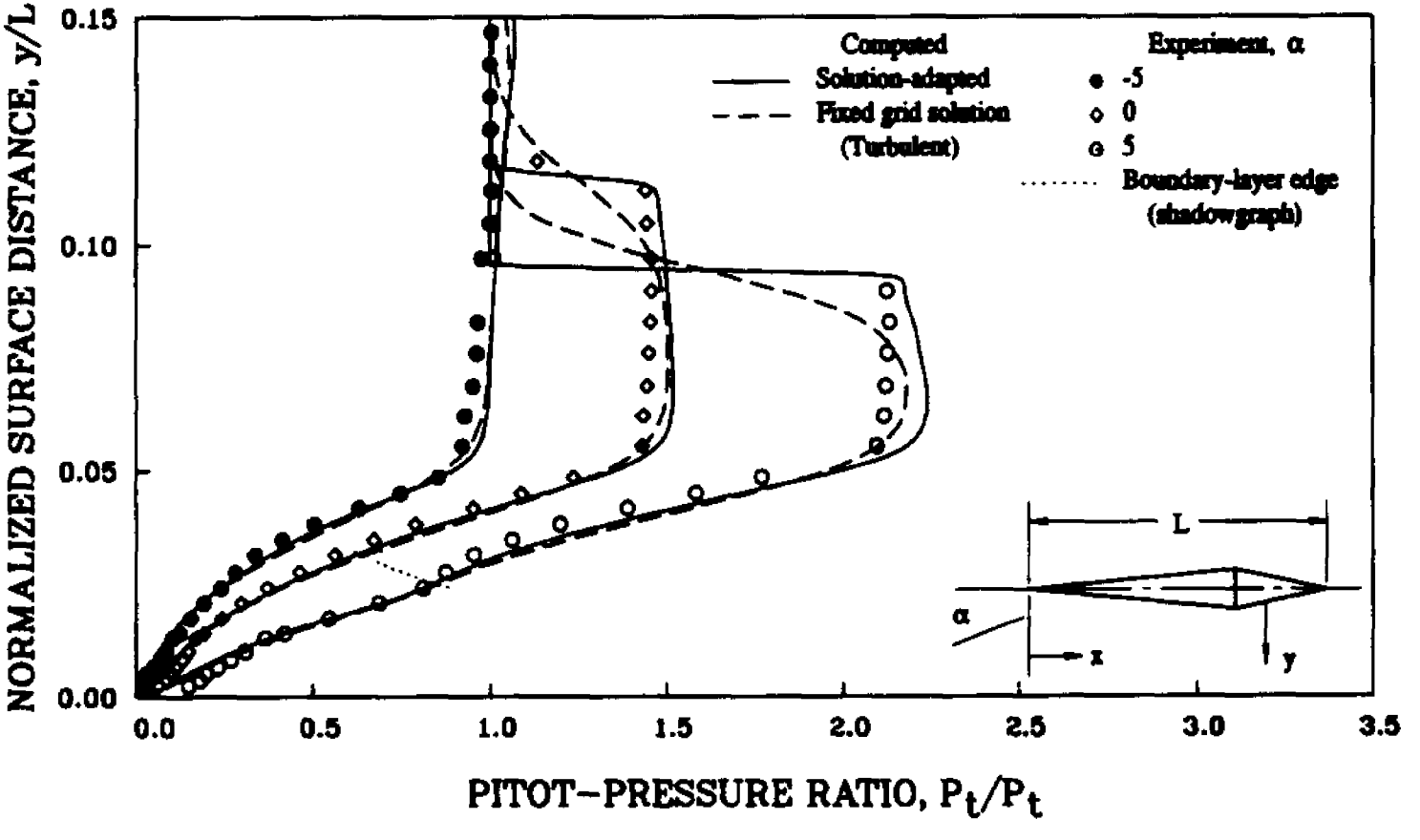


Figure 7.16: Comparison of pitot pressure profiles. (M_∞= 7.4, $Re_{\infty,L}$= 15×10^6, x/L= .8)

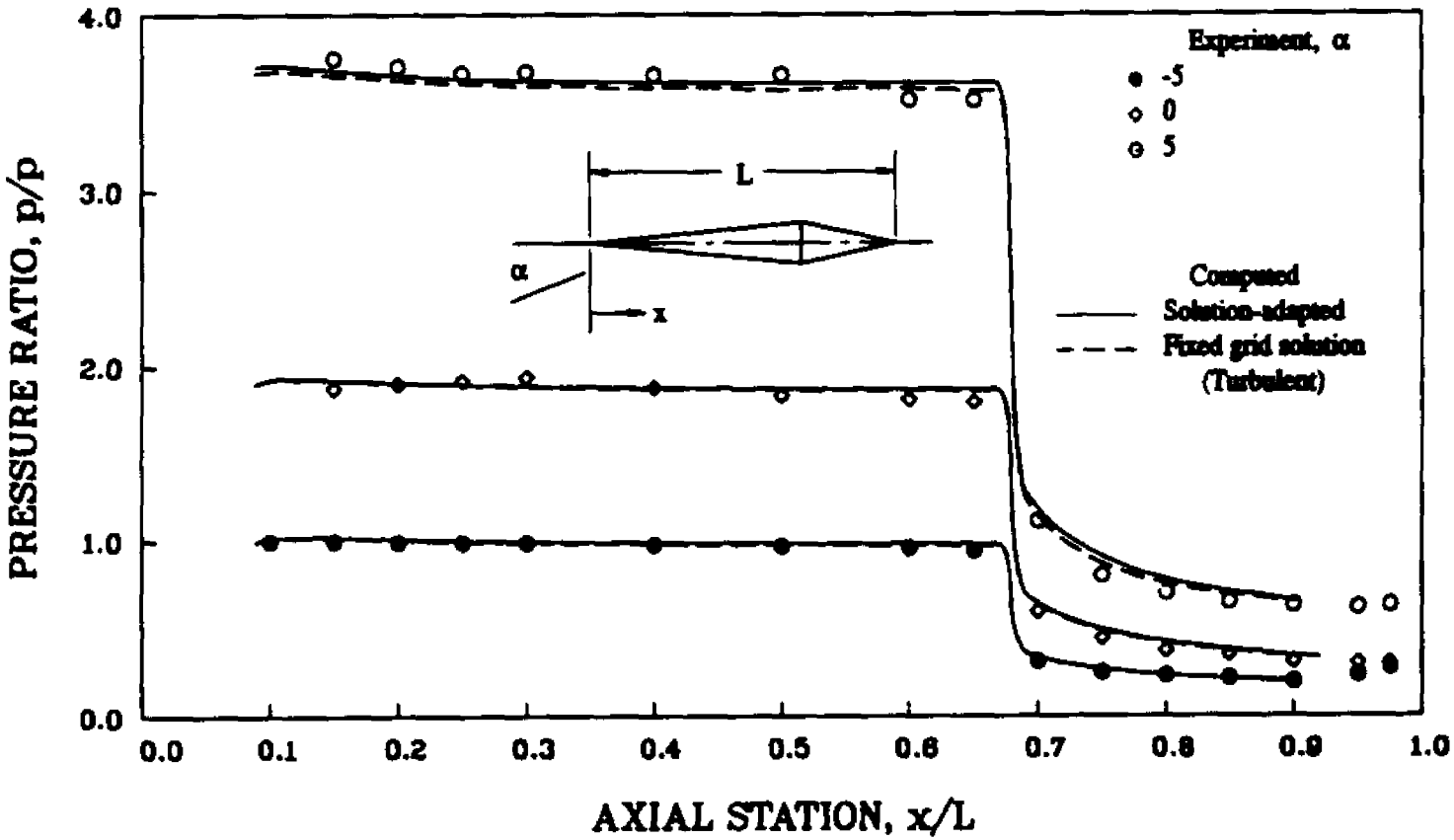


Figure 7.17: Comparison of symmetry plane surface pressure distributions. (M_∞= 7.4, $Re_{\infty,L}$= 15x10^6)

and a fixed grid show good agreement with experiment. The surface pressure remains constant along the forebody, then drops sharply at the start of the afterbody.

7.6 McDonnell Douglas Generic Option Vehicle

The solution-adaptive algorithm was also tested on the McDonnell Douglas Generic Option Vehicle at 0° angle of incidence. Flow-field conditions used are provided below.

$$M_\infty = 11.35 \qquad \tilde{T}_\infty = 60.2K \qquad \gamma = 1.4$$

$$Re/in = 9.14 \times 10^5 \qquad \tilde{T}_w = 311.1K \qquad Pr = 0.72$$

The unit Reynolds number, Re/in, is expressed per inch of the grid. The flow is turbulent and the eddy viscosity is computed using the algebraic model of Baldwin and Lomax, (1978). A perfect gas is assumed, and transition to fully turbulent flow is assumed to occur at $x = 1.5in$. Both solution-adapted and fixed grids consist of 65 points in each cross-flow direction. The surface grid and geometry are illustrated in Fig. 7.18.

Figure 7.19*a* illustrates the solution-adapted and grid-fitted grid on the forebody of the generic option. As expected, the grid is clustered in the important regions. Fig.7.19*b* shows the solution-adapted Mach contours, and Fig. 7.19*c* shows the corresponding contours on a fixed grid. The bow shock and the boundary layer region are well resolved in both the upper and lower planes in Fig. 7.19*b*. The outer grid boundary, due to grid-fitting, is accurately aligned with the bow shock. In Fig. 7.19*c* the fixed grid solution is clearly less resolved.

Figure 7.20 compares the fixed grid with solution-adapted grids obtained both with and without the grid-fitting algorithm at the maximum spanwise location for a chord distance of $14.68in$. Also shown are computed Mach contours. The fixed grid solution shows a smeared bow shock and the contours are wavy due

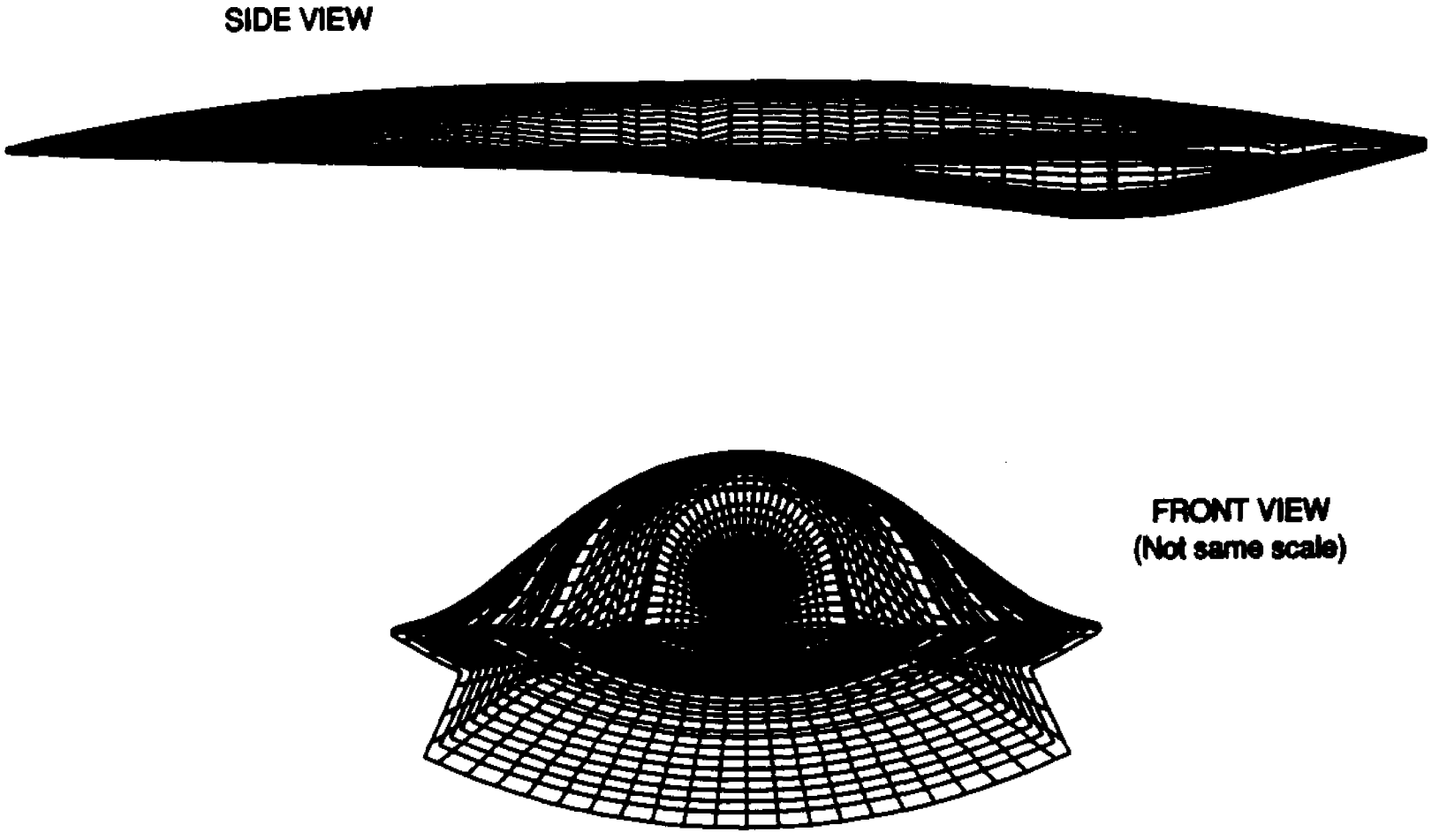


Figure 7.18: Surface grid for McDonnell Douglas Generic Option Vehicle.

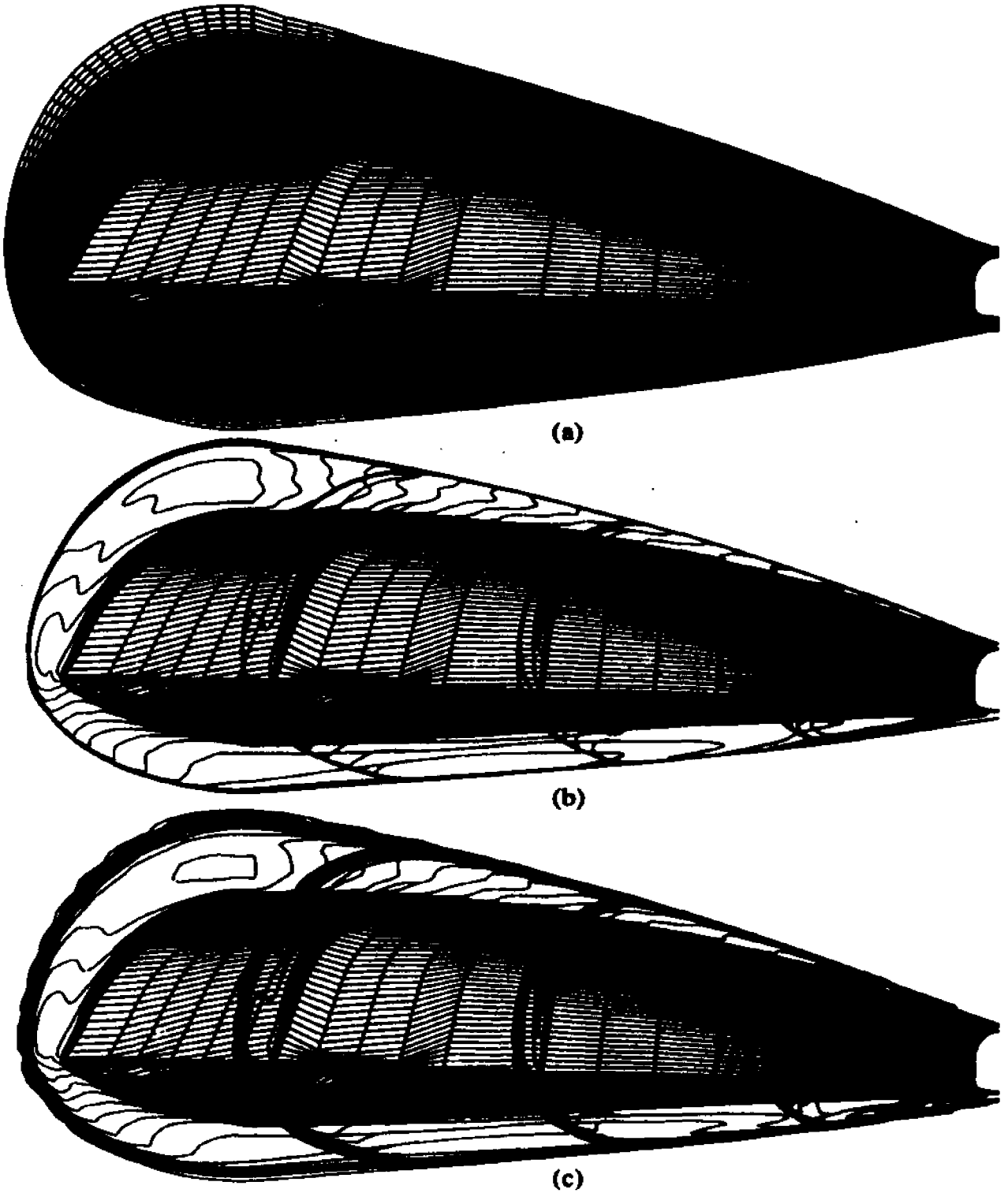


Figure 7.19: Illustration of solution-adapted and grid-fitted grid and Mach contours for McDonnell Douglas generic option vehicle, (a) grid (b) solution-adapted, grid-fitted (c) fixed grid predictions. (α= 0°, M_∞= 11.35, Re_∞= 9.14×10^5/in, 65x65 crossflow grid)

to poor resolution. For the solution-adapted case without grid-fitting, although the shock is better resolved, the oscillations in the Mach contours are still evident. This is attributable to the grid lines distorting outside the bow shock regions as discussed for the allbody results. The solution-adapted results with grid-fitting show significant improvements in the resolution of the bow shock region. The grid is smooth and reasonably orthogonal but provides the necessary clustering where required.

Figure 7.21 compares the resulting solution-adaptive computational grid with a fixed grid for the afterbody of the option vehicle ($x = 17 - 28in$). Grid surfaces are plotted in both upper and lower symmetry planes. A crossflow grid plane is also shown at $x = 28in$. The locations of the ramp shocks are marked by grid clustering regions in the lower symmetry plane for the solution-adapted case.

Figure 7.22 compares Mach number contours at three different x-locations between $x = 17$ and $28in$ computed using a fixed computational grid with those computed using a solution-adapted grid for which the grid-fitting option was not implemented and adaptations were performed only in the radial direction. Mach contours in the lower symmetry plane of Fig. 7.22*b*, showing the interaction of the compression shocks, are well resolved by the use of the adaptive grid. The corresponding fixed grid contours exhibit considerable smearing of the bow and compression shocks due to lack of the necessary grid density in these regions. At the lower surface of the upstream-most contour section of Fig. 7.22*b*, the first compression shock can be seen just below the outer edge of the boundary layer. At the second cross-section this shock wave has moved away from the body surface and is about midway between the body surface and the bow shock. The shock due to the second compression surface interacts with the bow shock at the location of the third set of cross-sectional Mach contours. Also enhanced by the solution-adaptive grid code is the shock interaction at the corner of the wing section of

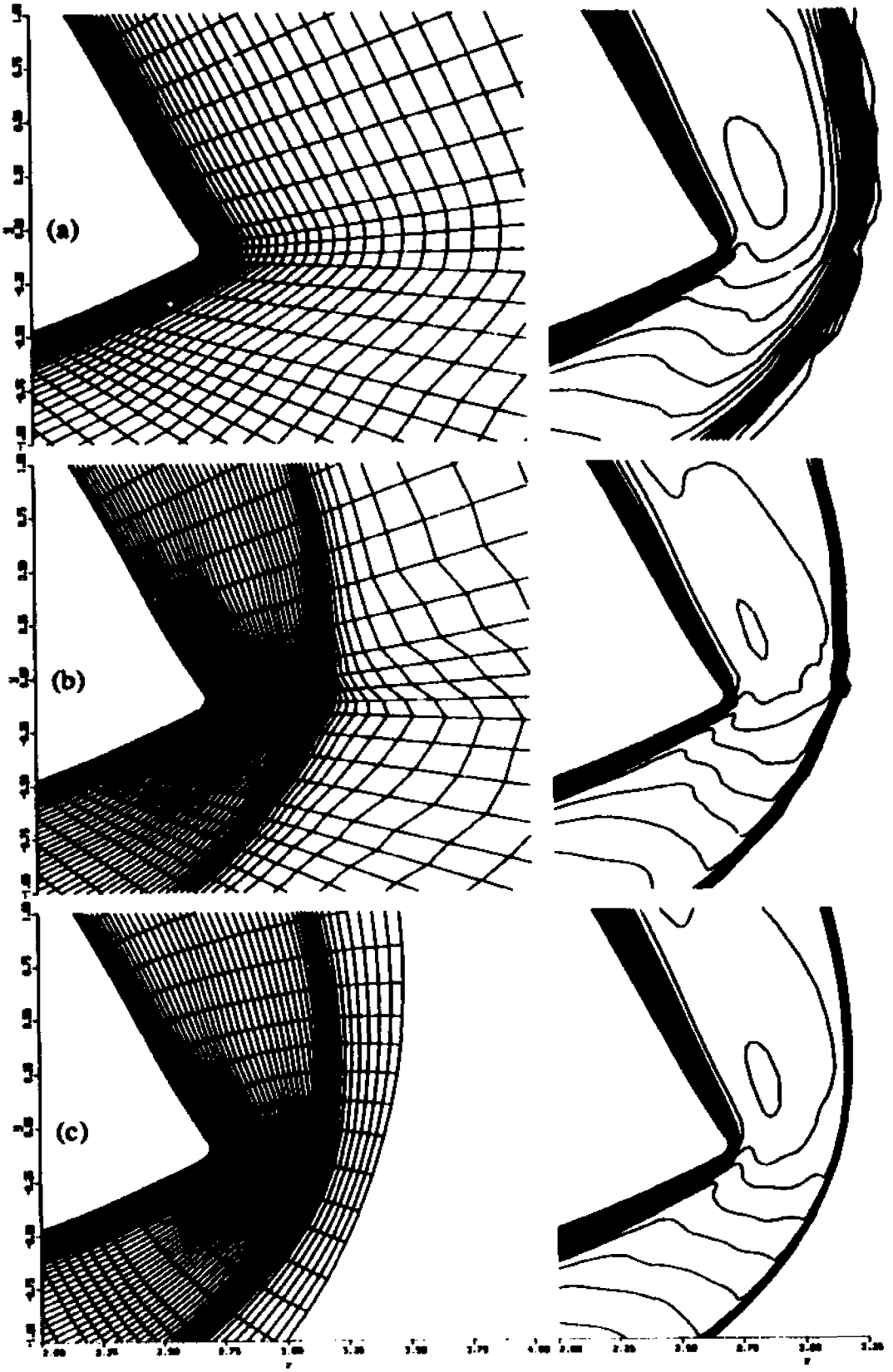


Figure 7.20: Comparison of grids and Mach number contours near maximum spanwise location. (a) fixed grid, (b) adapted, w/o grid-fitting, (c) adapted and grid fitted. (α= 0°, M_∞= 11.35, Re_∞= 9.14x10^5/in, x= 14.68in, 65x65 grid)

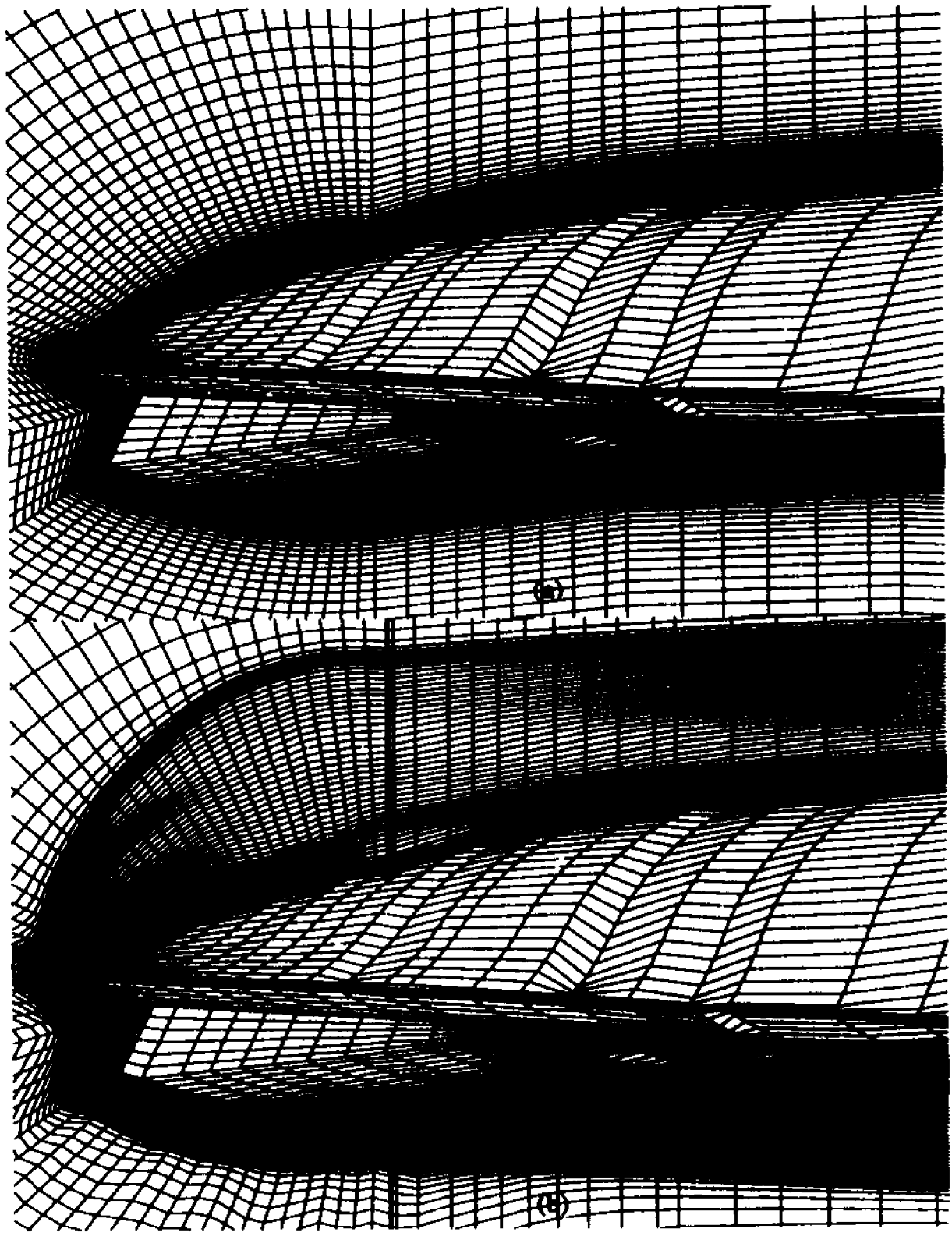
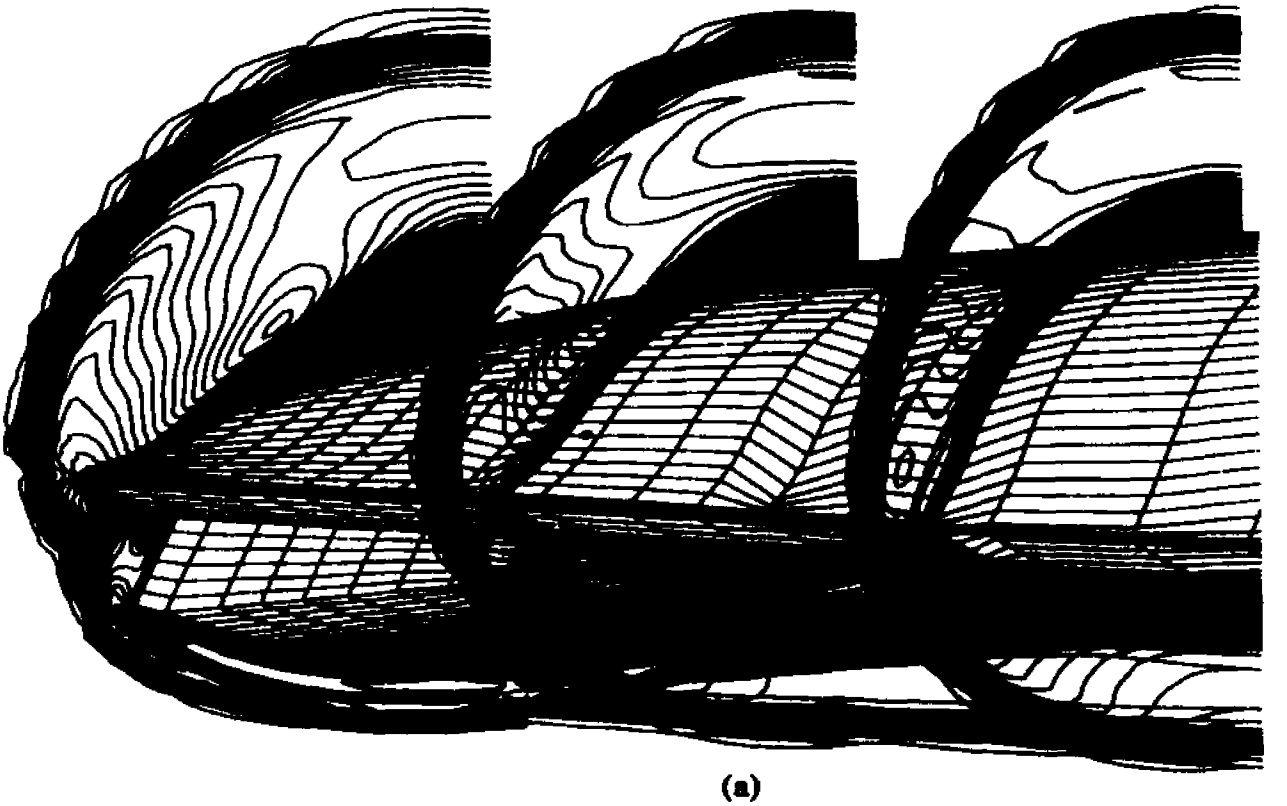


Figure 7.21: Comparison of computational grids at x=17-28in. for McDonnell Douglas generic option vehicle. (a) fixed grid, (b) solution-adapted. (α= 0°, M_∞= 11.35, Re_∞= 9.14×10^5/in, 65x65 crossflow grid)



(a)

(b)

Figure 7.22: Comparison of Mach number contours at x=17-28in. for McDonnell Douglas generic option vehicle. (a) fixed grid, (b) solution-adapted. (α= 0°, M_∞= 11.35, Re_∞= 9.14×10^5/in, 65x65 crossflow grid)

the third Mach contour section. The expansion fan resulting from the interaction of the two ramp shocks is clearly resolved in the solution adapted results. These features of the flow are not as easily recognizable in the Mach contours of Fig. 7.22*a*.

CHAPTER 8

Application to Chemical Nonequilibrium Flows

An important characteristic of the hypersonic flow regime is the presence of viscous dissipation sufficient to cause temperatures high enough to dissociate the gas. In this chapter, the solution-adaptive grid algorithm is applied for predicting the chemically reacting flow of air over a 5° half-angle blunt cone at 5° angle of incidence. In the chemistry model presented in this chapter, the fluid medium is considered to be a chemically reacting mixture of thermally perfect, but calorically imperfect gases in thermal equilibrium.

8.1 Overview of Chemistry Model

In the solution of the flow of a chemically reacting mixture of gases, the chemical reactions involved must also be considered in addition to the fluid dynamics. The present solution method incorporates a loose coupling of the chemistry equations with those of the fluid dynamics and was developed be Buelow et al., (1990). In this approach, instead of solving the fluid dynamic equations together with the species continuity equations as a single set, the flow equations and the species transport equations are solved successively in an iterative framework. In the strongly coupled approach [Molvik, (1989); Prabhu et al., (1987)], the chemical species continuity equations are solved together, with the conservation equations of mass, momentum and energy as a large set and, as a result, the influence of the chemistry on the flow is resolved in a single application of the equation solver. However, for large numbers of species, the equation set which results from the use of the strongly coupled approach can become large and its solution becomes increasingly more expensive to compute.

The chemistry model of Buelow, (1990) used in the present calculations consists of molecular oxygen (O_2), atomic oxygen (O), molecular nitogen (N_2), atomic nitrogen (N), nitric oxide (NO) and nitric oxide ion (NO^+). The following reactions between these constituent reactants are considered.

$$(1) \qquad O_2 + M_1 \rightleftharpoons 2O + M_1$$

$$(2) \qquad N_2 + M_2 \rightleftharpoons 2N + M_2$$

$$(3) \qquad N_2 + N \rightleftharpoons 2N + N$$

$$(4) \qquad NO + M_3 \rightleftharpoons N + O + M_3$$

$$(5) \qquad NO + O \rightleftharpoons O_2 + N$$

$$(6) \qquad N_2 + O \rightleftharpoons NO + N$$

$$(7) \qquad N + O \rightleftharpoons NO^+ + e^-$$

where $M_{1,2,3}$ are catalytic third bodies. This model contains six species ($n = 6$), seven reactions and ten reactants. The mass production rates are calculated using the expressions found in Blottner et al. (1971).

Another advantage of using the loosely coupled approach in the present work, is that only minor modifications to the PNS equations (as introduced in Chapter 2), are necessary. The effects of species mass diffusion is introduced into the energy equation as a component added to the heat flux terms (q_x, q_y and q_z) and has the following form

$$\rho \sum_{s=1}^{n} c_s V_s (h_s - h_n) \qquad (s = 1, 2, \ldots, n)$$

where c_s, V_s and h_s is the mass fraction, diffusion velocity and enthalpy, respectively, of species s. The species continuity equations are simplified by the assumption of global continuity and through the use of Fick's Law for mass diffusion, [Buelow et al, (1990)]. The parabolized Navier-Stokes assumptions of Chapter 2 are also applied, eliminating the unsteady terms and neglecting the streamwise diffusion. The final form of the species continuity equations are written in the curvilinear coordinate system as

$$\rho \hat{U} \frac{\partial c_s}{\partial \xi} + \rho \hat{V} \frac{\partial c_s}{\partial \eta} + \rho \hat{W} \frac{\partial c_s}{\partial \zeta} - \frac{\partial}{\partial \eta}\left[A_{\eta\eta} \frac{\partial c_s}{\partial \eta} + A_{\eta\zeta} \frac{\partial c_s}{\partial \zeta}\right] - \frac{\partial}{\partial \zeta}\left[A_{\zeta\zeta} \frac{\partial c_s}{\partial \zeta} + A_{\zeta\eta} \frac{\partial c_s}{\partial \eta}\right] = \frac{\dot{p}_s}{J} \quad (s = 1, 2, 3, \ldots, n)$$

where

$$\hat{U} = \left(\frac{\xi_x}{J}\right)u + \left(\frac{\xi_y}{J}\right)v + \left(\frac{\xi_z}{J}\right)w$$

$$\hat{V} = \left(\frac{\eta_x}{J}\right)u + \left(\frac{\eta_y}{J}\right)v + \left(\frac{\eta_z}{J}\right)w$$

$$\hat{W} = \left(\frac{\zeta_x}{J}\right)u + \left(\frac{\zeta_y}{J}\right)v + \left(\frac{\zeta_z}{J}\right)w$$

$$A_{\eta\eta} = \rho D\left[\left(\frac{\eta_x}{J}\right)^2 + \left(\frac{\eta_y}{J}\right)^2 + \left(\frac{\eta_z}{J}\right)^2\right]$$

$$A_{\zeta\zeta} = \rho D\left[\left(\frac{\zeta_x}{J}\right)^2 + \left(\frac{\zeta_y}{J}\right)^2 + \left(\frac{\zeta_z}{J}\right)^2\right]$$

$$A_{\zeta\eta} = A_{\eta\zeta} = \rho D\left[\left(\frac{\eta_z}{J}\right)\left(\frac{\zeta_x}{J}\right) + \left(\frac{\eta_z}{J}\right)\left(\frac{\zeta_y}{J}\right) + \left(\frac{\eta_z}{J}\right)\left(\frac{\zeta_z}{J}\right)\right]$$

As before, c_s is the mass fraction of species s, D is the kinematic binary diffusion coefficient, $\dot{p}_s$ is the species production term and the Jacobian and metric terms are identical in form to those presented in Chapter 2.

The flow-chemistry coupling is enhanced through the use of a Newton sub-iteration procedure performed on the flow equations. This procedure can best be described by first re-writing Eq. (2.7) at the $n+1$ marching station as

$$\hat{\mathbf{E}}_{\xi}^{n+1} + \hat{\mathbf{F}}_{\eta}^{n+1} + \hat{\mathbf{G}}_{\zeta}^{n+1} = 0 \tag{8.1}$$

The fluxes are then linearized about the $n+1$ marching station in the following manner

$$(\hat{\mathbf{E}}^{n+1})^{i+1} = (\hat{\mathbf{E}}^{n+1})^{i} + \left(\frac{\partial \hat{\mathbf{E}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}$$

$$(\hat{\mathbf{F}}^{n+1})^{i+1} = (\hat{\mathbf{F}}^{n+1})^{i} + \left(\frac{\partial \hat{\mathbf{F}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}$$

$$(\hat{\mathbf{G}}^{n+1})^{i+1} = (\hat{\mathbf{G}}^{n+1})^{i} + \left(\frac{\partial \hat{\mathbf{G}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}$$

where, $\mathbf{U}$ is the vector of conserved variables and

$$\Delta^{i}\mathbf{U} = (\mathbf{U}^{n+1})^{i+1} - (\mathbf{U}^{n+1})^{i}$$

Upon substitution of the linearized fluxes into Eq. (8.1), the PNS equations are then written as

$$\begin{aligned} &\frac{\partial}{\partial \xi}\left[\left(\frac{\partial \hat{\mathbf{E}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}\right] + \frac{\partial}{\partial \eta}\left[\left(\frac{\partial \hat{\mathbf{F}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}\right] \\ &\quad + \frac{\partial}{\partial \zeta}\left[\left(\frac{\partial \hat{\mathbf{G}}^{n+1}}{\partial \mathbf{U}}\right)^{i} \Delta^{i}\mathbf{U}\right] = -[\hat{\mathbf{E}}_{\xi}^{n+1} + \hat{\mathbf{F}}_{\eta}^{n+1} + \hat{\mathbf{G}}_{\zeta}^{n+1}]^{i} \end{aligned} \tag{8.2}$$

As the sub-iterations continue, the terms on the left-hand side go to zero, resulting in the solution to the PNS equations at the $n+1$ marching station [Buelow, (1990)].

8.2 Solution Procedure

The solution of the equations governing the flow of a mixture of chemically reacting gases using the solution-adaptive grid algorithm is now described. Using the numerical integration techniques described in Chapter 2, the solution for the flow conservation equations are advanced to the $n+1$ marching station assuming frozen chemistry. Using the fluid properties at $n+1$ (density and velocities), the species continuity equations are marched from the nth marching station to the $n+1$ marching station, yielding the species mass fractions at the $n+1$ marching station. The enthalpies (J/kg) and specific heats $(J/kg \cdot K)$ of the species are then obtained from the following relations

$$\tilde{h}_s = \tilde{T} A_s(\tilde{T}) + \tilde{h}_s^\circ$$

$$\tilde{C}_{p,s} = \frac{d\tilde{h}_s}{d\tilde{T}} = B_s(\tilde{T})$$

where, A and B are interpolated from the tables of Blottner, (1971) using cubic splines and $\tilde{h}_s^\circ$ is the enthalpy of formation (J/kg) of species s. The enthalpy and specific heat of the locally frozen mixture are then computed from the following relations.

$$\tilde{h}_f = \sum_{s=1}^{n} c_s \tilde{h}_s$$

$$\tilde{C}_{p_f} = \sum_{s=1}^{n} c_s \frac{d\tilde{h}_s}{d\tilde{T}} = \sum_{s=1}^{n} c_s \tilde{C}_{p,s}$$

With the enthalpy of the fluid known, $\tilde{h} = f(\tilde{T}) = \sum c_s \tilde{h}_s(\tilde{T})$, a Newton-Rasphson fixed point iteration of the following form is performed to obtain the temperature, i.e,

$$\tilde{T}^{k+1} = \tilde{T}^k - \frac{\sum c_s \tilde{h}_s(\tilde{T}^k) - \tilde{h}}{\sum c_s \tilde{C}_{p,s}(\tilde{T}^k)}$$

where the summation is taken over all the species. Once the temperature is known the pressure can be determined from Dalton's Law of partial pressures for a mixture of thermally perfect gases. The fluid viscosity $(kg/m \cdot s)$ of each

species is calculated from the data of Blottner (1971). The thermal conductivity ($W/m \cdot K$) of each species is computed using Eucken's semi-emperical formula (See, for example, Bird et al., 1960). The viscosity and thermal conductivity of the mixture are then calculated using Wilke's mixing rule [Wilke, (1950)]. The reader is referred to the literature for the exact form of these relations. The binary Lewis numbers, Le, are assumed to be the same for each species; thus the binary diffusion coefficient D (m^2/s) is computed from the definition ($D = \tilde{k}\mathrm{Le}/\tilde{\rho}\tilde{C}_p$).

After both the flow and chemistry have been advanced to the $n+1$ marching station and flow properties are determined in the manner described above, the next step is to adapt the grid based on the solution at the $n+1$ station and at the first level of the Newton iteration procedure. The grid is adapted to the solution in the same manner as described in the last chapter, sweeping each coordinate line in both crossflow direction until all grid points have been re-distributed based on initial solution error estimation.

After the grid is adapted, the flow equations are re-computed at the $n+1$ marching station using the upstream solution at the nth station. This is followed by a solution of the chemistry using the refined density and velocity values obtained on the adapted grid at $n+1$. The fluid properties at the $n+1$ station are then advanced from iteration i to $i+1$ with Eq. (8.2) using the species mass fractions and thermodynamic and transport properties at sub-iteration i and station $n+1$. The chemistry is then iterated to level $i+1$ using the density values at marching station $n+1$ and iteration level $i+1$. This process is continued until a desired level of convergence is obtained. Besides reducing error associated with the loosely coupled fluids-chemistry approach, the Newton sub-iteration procedure of Eq. (8.2) also reduces errors due to the flux linearizations and approximate factorization.

Once a desired level of convergence is obtained between the chemistry and the flow, the grid-fitting algorithm is employed to determine the position of the outer grid line at the next station and the adapted grid plane is projected downstream in preparation for the next step.

8.3 Flow over a Blunt Cone

The solution-adaptive PNS solver is used to compute the laminar flow of chemically reacting air over a 5° half-angle blunt nosed cone. The flow conditions are given below.

$$\tilde{V}_\infty = 4533 m/s \qquad (c_{N_2})_\infty = .7371$$

$$\tilde{p}_\infty = 1272 N/m^2 \qquad (c_{O_2})_\infty = .2629$$

$$\tilde{T}_\infty = 702 K \qquad (c_N)_\infty = (c_O)_\infty = (c_{NO})_\infty = (c_{NO^+})_\infty = 0$$

$$\tilde{T}_w = 1216 K \qquad \alpha = 5^\circ$$

These conditions correspond to an altitude of approximately 100 kilometers. The appropriate non-dimensional parameters are $M_\infty = 15$, $Re_\infty = 1.33 \times 10^5$ and $Pr_\infty = 0.72$. The computational grid consists of 65 grid cells in the circumferential direction and 64 cells in the radial direction for both fixed and solution-adapted computations. An initial blunt body solution plane of data at $x = 7.5 \times 10^{-3} m$ was obtained from the Navier-Stokes code of Molvik, (1989). The solution-adaptive marching procedure was employed to a distance of $1.016 m$ $(40 in.)$ downstream using a marching stepsize of $(0.1 in)$.

Figure 8.1 illustrates the resulting solution-adapted and grid-fitted computational grid. Grid clustering is observed at the bow shock locations in the symmetry planes and cross-flow planes. The outer grid surface is also observed to be sufficiently smooth. Its computed location adequately minimizes the number of grid points in the freestream. The initial plane of data was not adapted or grid-fitted. Thus, there exists a short transition region near the blunt nose where the grid

comes into equilibrium with the solution. This section of the solution-adapted grid is shown in Fig. 8.2. Also shown is the outer grid surface in this same region as determined by the grid-fitting scheme. During this transition, the torsional parameter, λ was given a value of 10^{-2}, and was ramped down to 10^{-4} over 100 marching stations. This procedure prevented the grid from changing too rapidly while the grid was changing from the initially fixed grid to a fully adapted grid; this gradual change prevented the calculations from becoming unstable. The intial grid plane consists of grid points exponentially stretched from the boundary in the ζ-direction using the Robert's stretching function and uniform spacing in the circumferential direction.

The maximum and minimum grid spacings in the ζ-direction where given values of $(\Delta s_{MAX})_\zeta/\Delta s_{ave} = 2.5$, $(\Delta s_{MIN})_\zeta/\Delta s_{ave} = 0.6$. Maximum and minimum grid spacings in the η-direction where $(\Delta s_{MAX})_\eta/\Delta s_{ave} = 1.0$, $(\Delta s_{MIN})_\eta/\Delta s_{ave} = 0.9$. The straightness parameter used for the radial adaptation was $C_\zeta = 0.6$ and the straghtness parameter used for circumferential adaptations was given a value of $C_\eta = 0.5$. The radial torsion parameter was given a value of $\lambda_\zeta = 10^{-4}$ (with the modification discussed above while adapting close to the nose) and the circumferential torsion parameter $\lambda_\eta = 5 \times 10.^{-3}$

Figure 8.3 shows Mach contours resulting from the solution-adapted computations. Figure 8.4 show Mach contours resulting from the fixed grid computations. Significant improvements are detected in the resolution of the bow shock for the solution-adapted case. Improvements in resolution of the bulge in the boundary layer on the leeward side can also detected.

In Fig. 8.5, computed atomic oxygen contours are plotted at several marching planes. Figure 8.6 illustrates a similar plot resulting from the fixed grid computations. Considerably more dissociation occurs on the windward side than on the leeward side. As the flow proceeds downstream, the atomic oxygen tends to

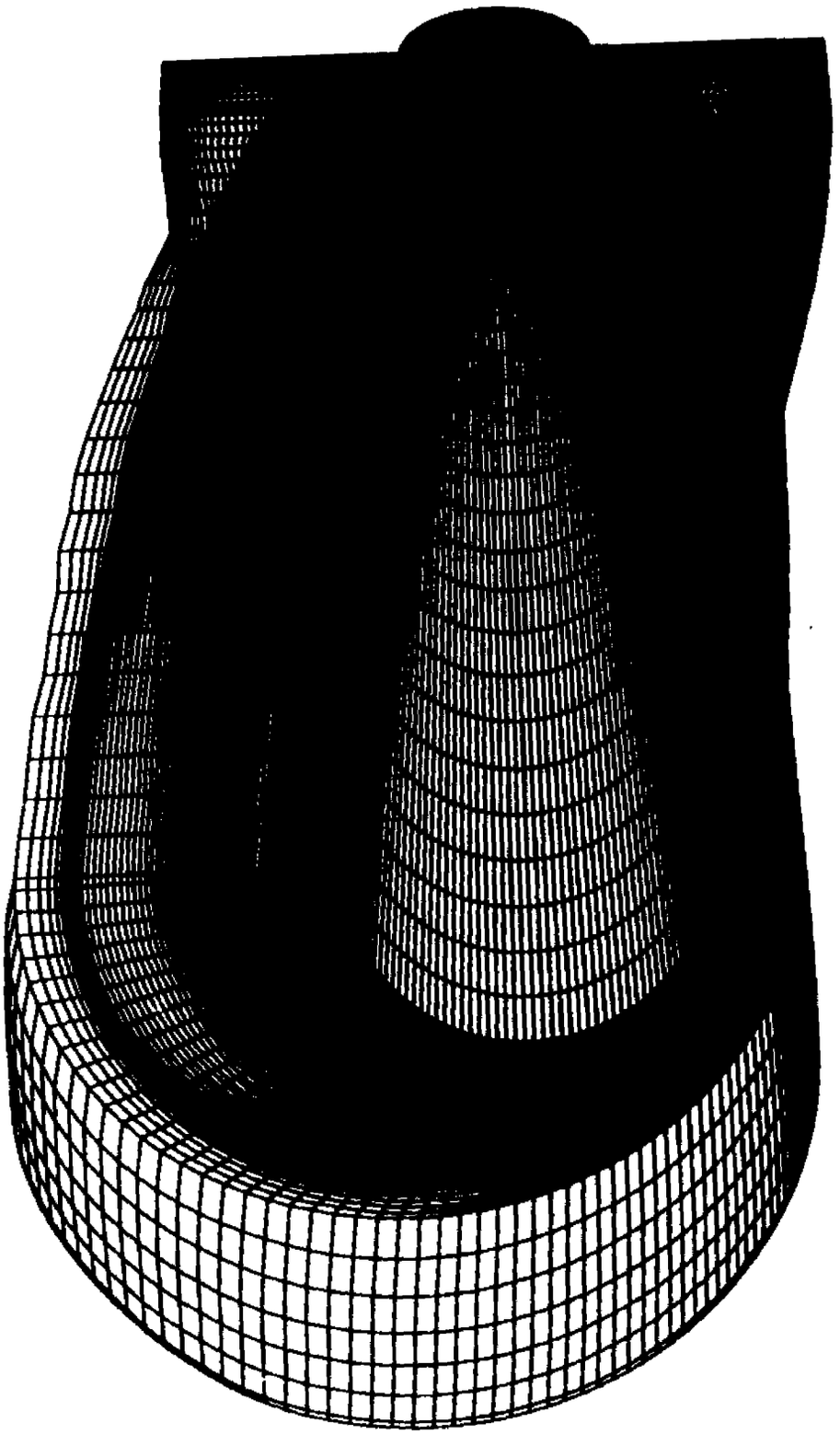


Figure 8.1: Solution-adapted and grid-fitted grid. (α= 5.° M_∞= 15, $Re_{\infty L}$= 1.33×10^5, T_∞= 227K, 65x65 grid)

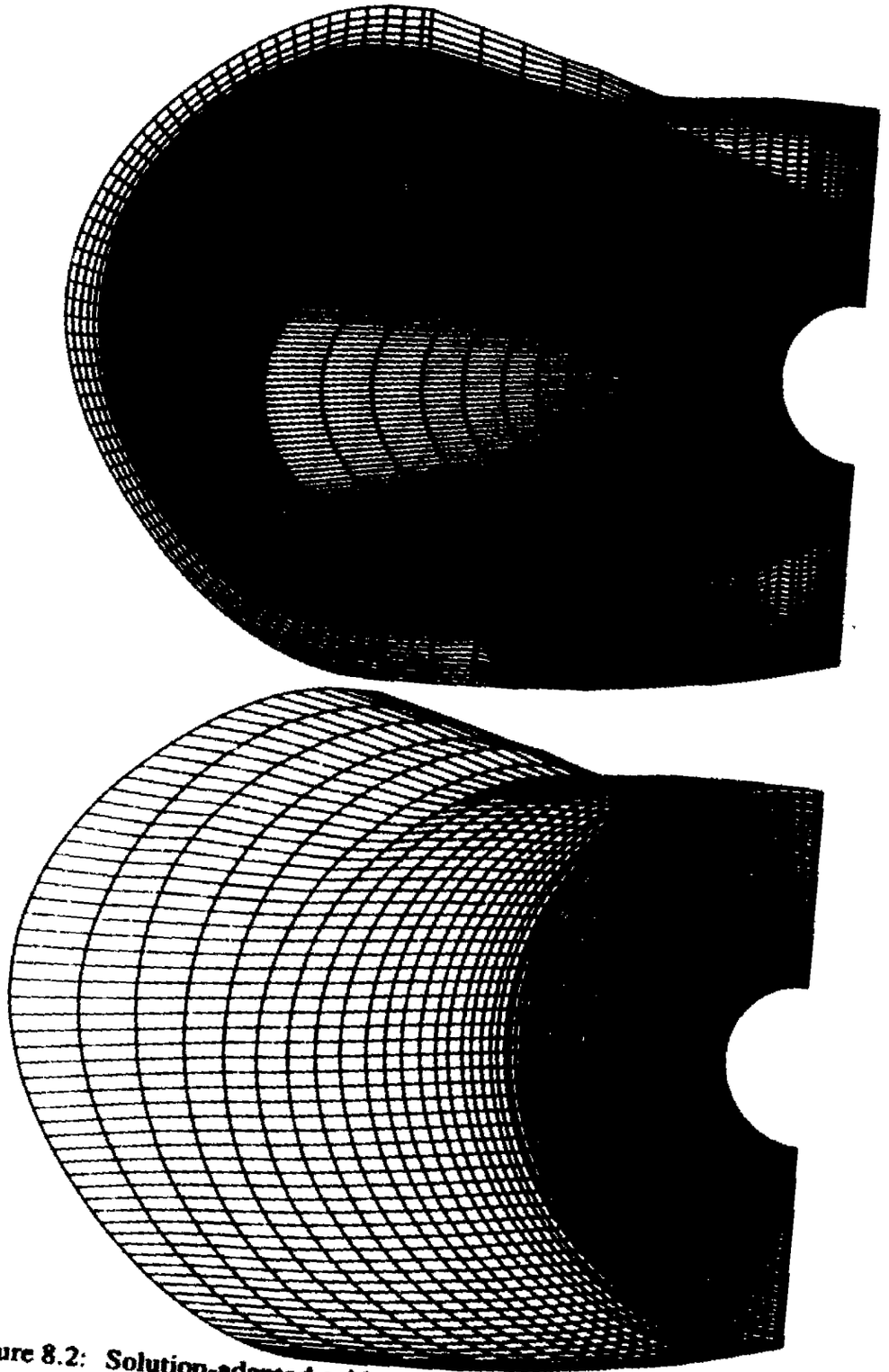


Figure 8.2: Solution-adapted grid and outer grid surface for blunt cone near start of marching procedure showing transition from fixed to adapted grid.

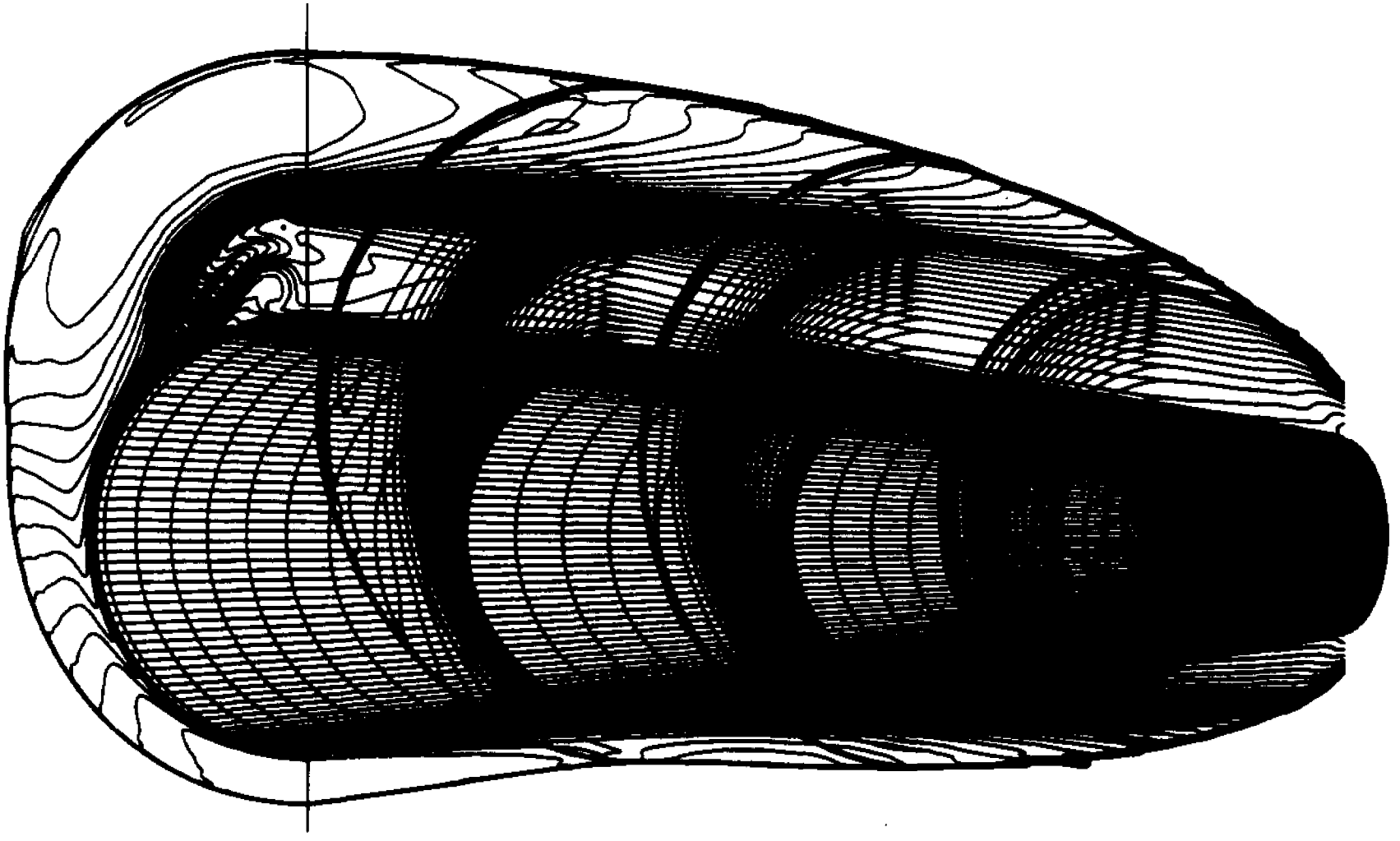


Figure 8.3: Mach number contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 ADAPTED grid)

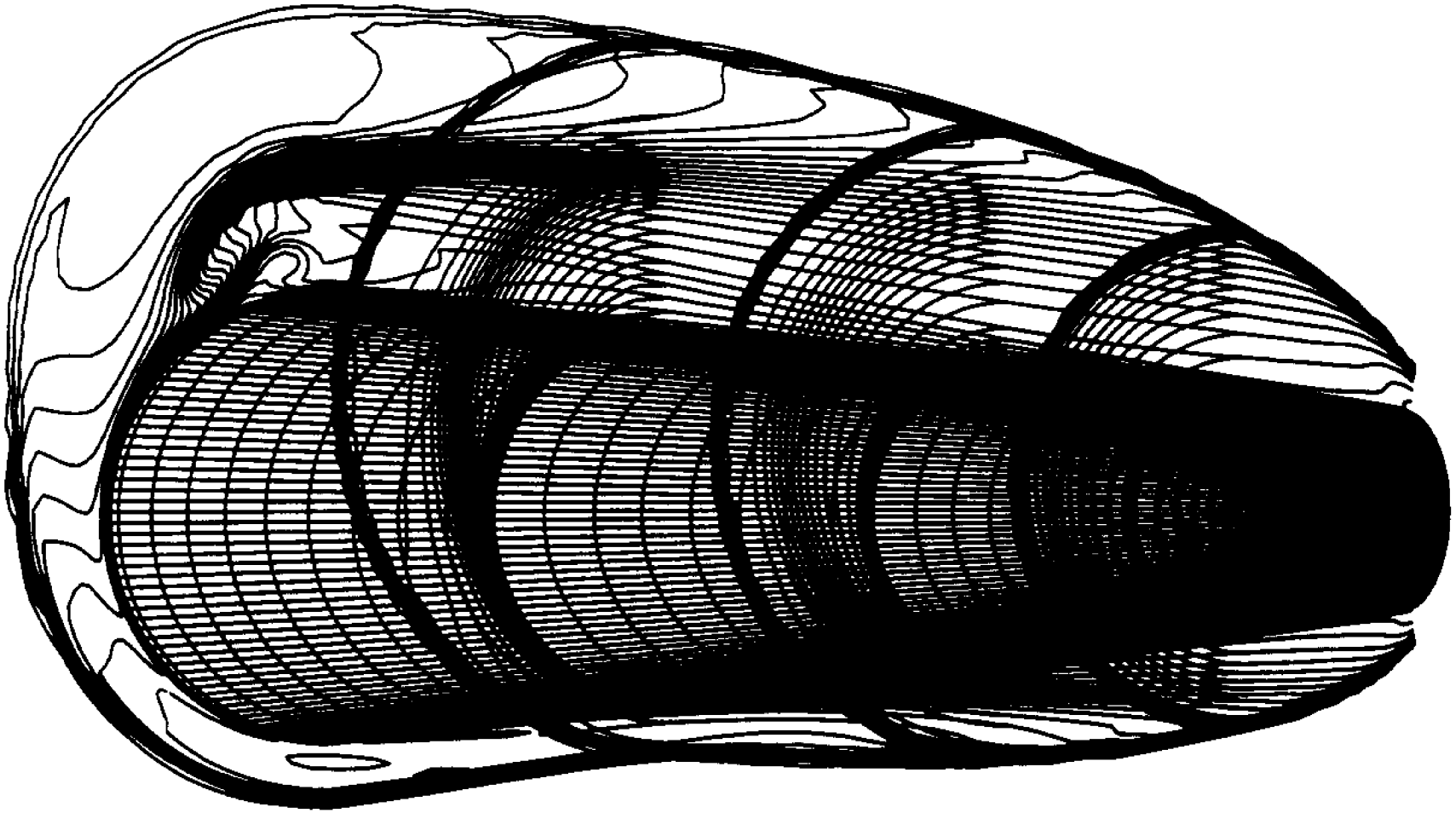


Figure 8.4: Mach number contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 FIXED grid)

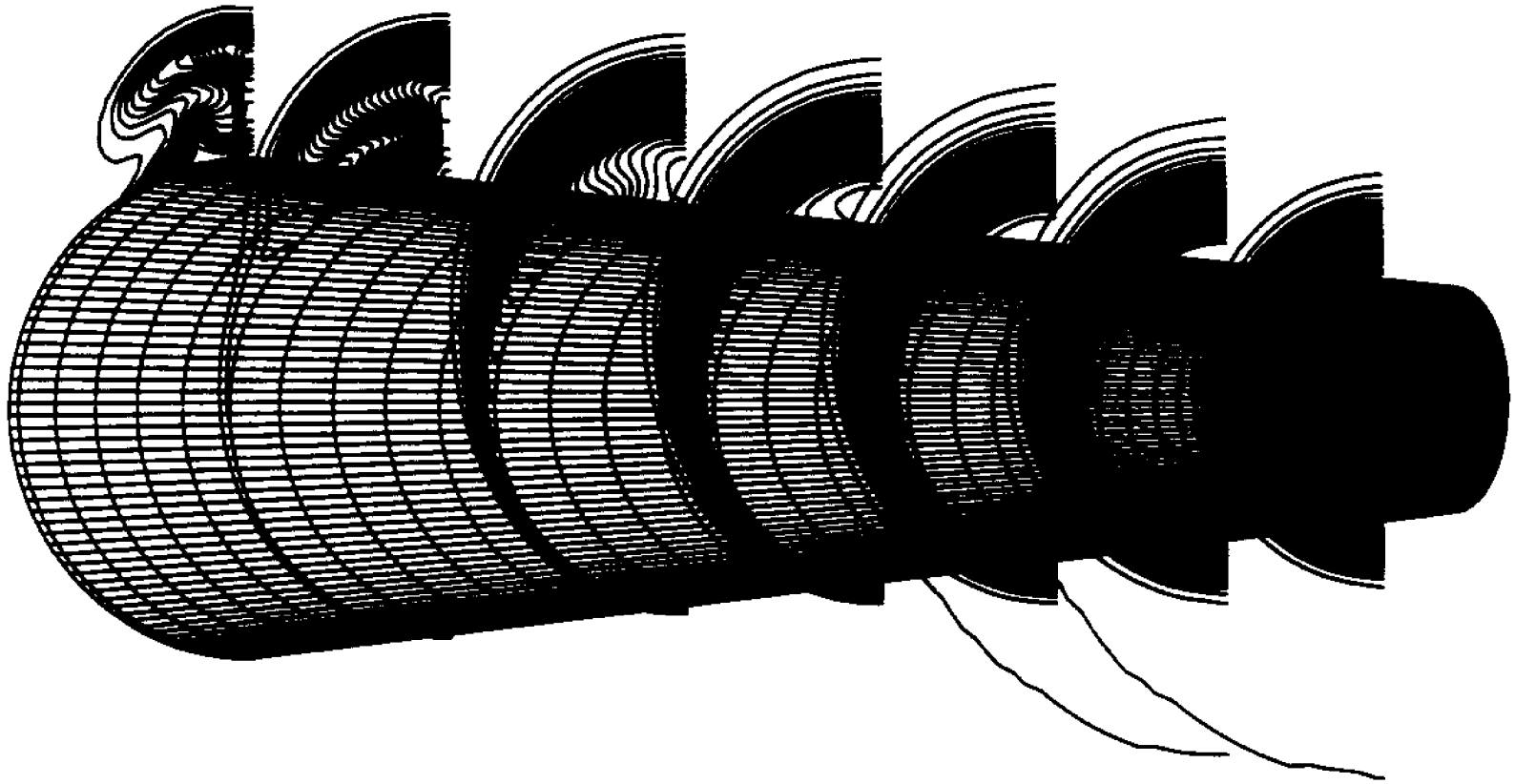


Figure 8.5: Atomic oxygen contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 ADAPTED grid)

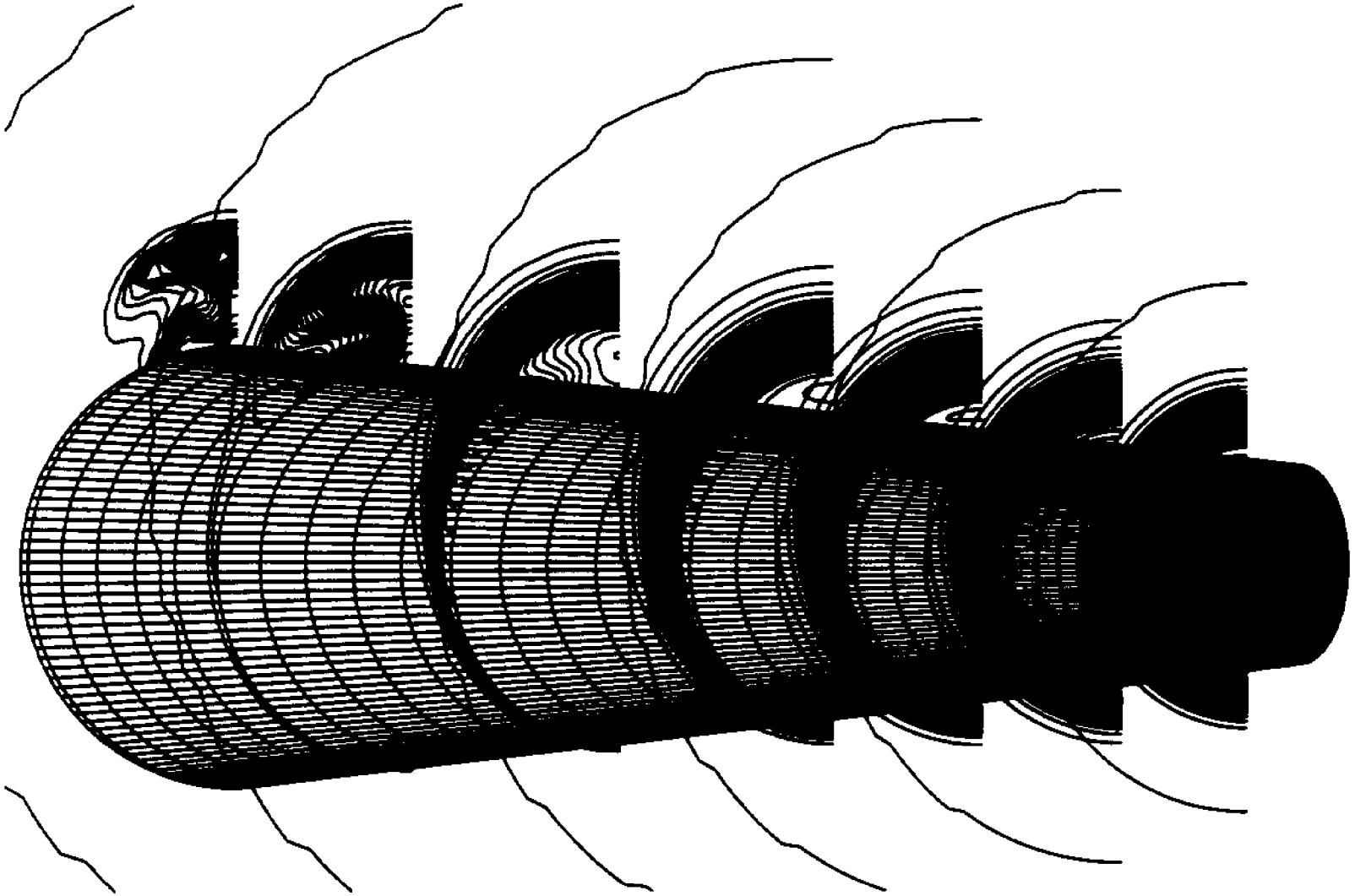


Figure 8.6: Atomic oxygen contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 FIXED grid)

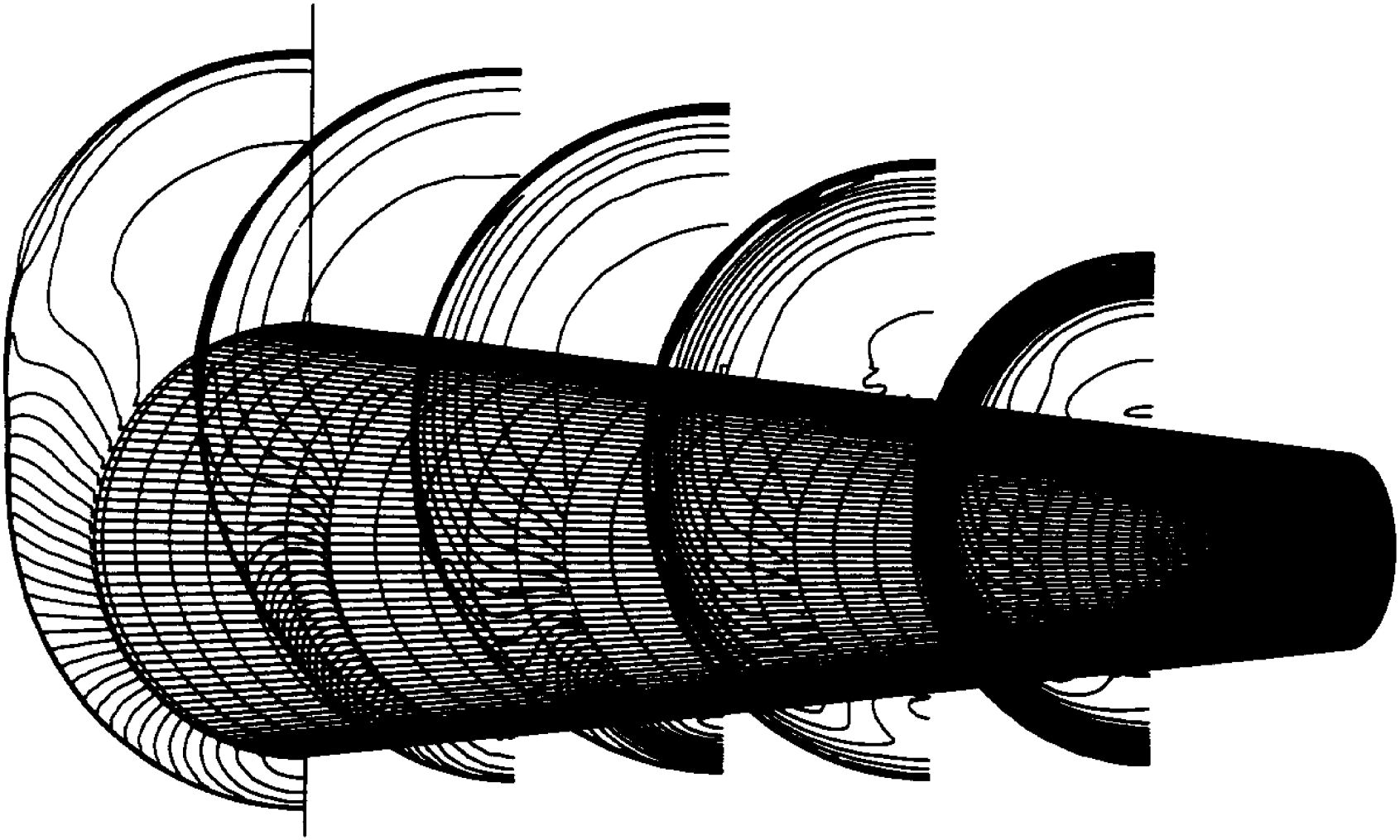


Figure 8.7: Pressure contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 ADAPTED grid)

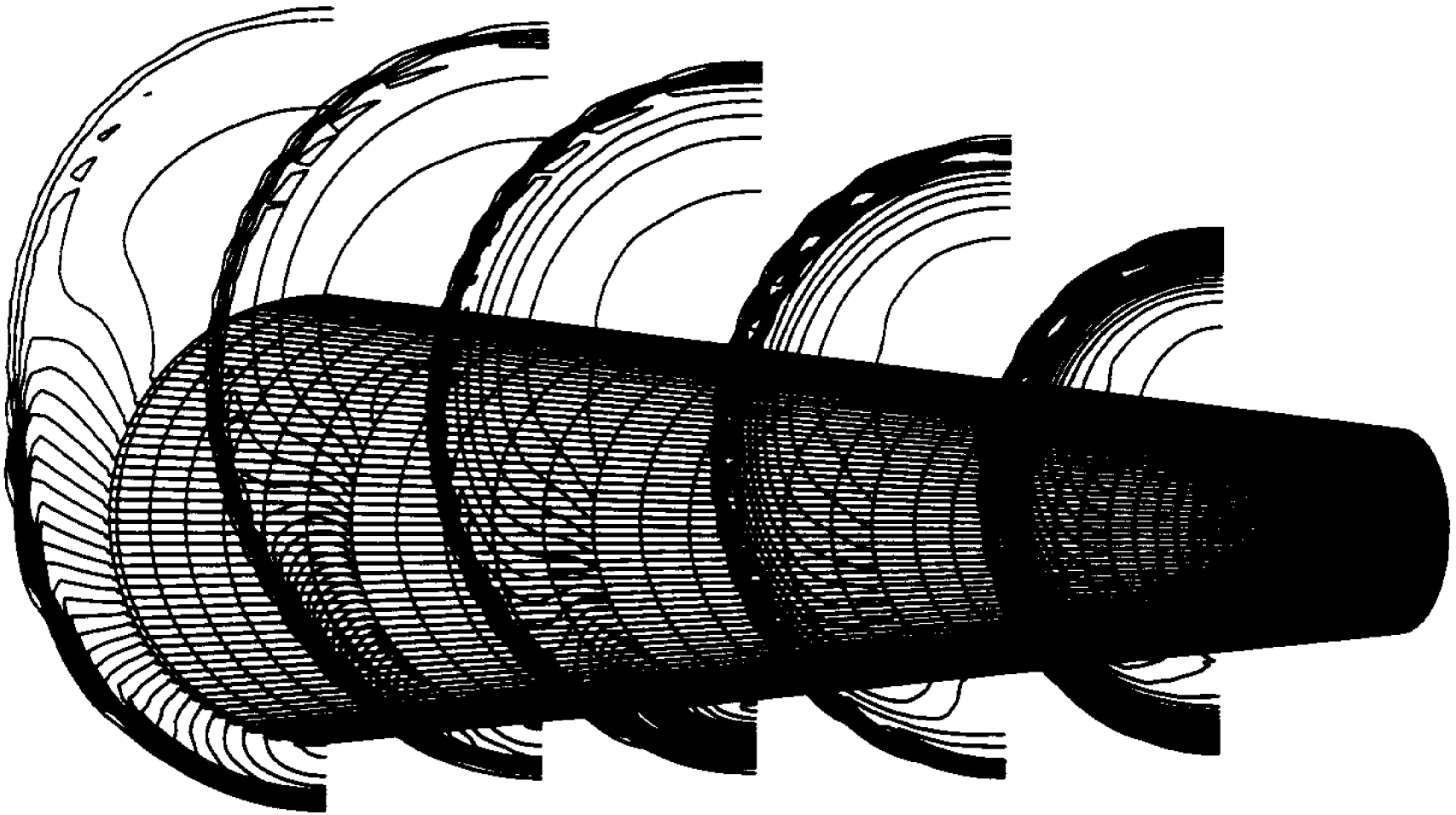


Figure 8.8: Pressure contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x65 FIXED grid)

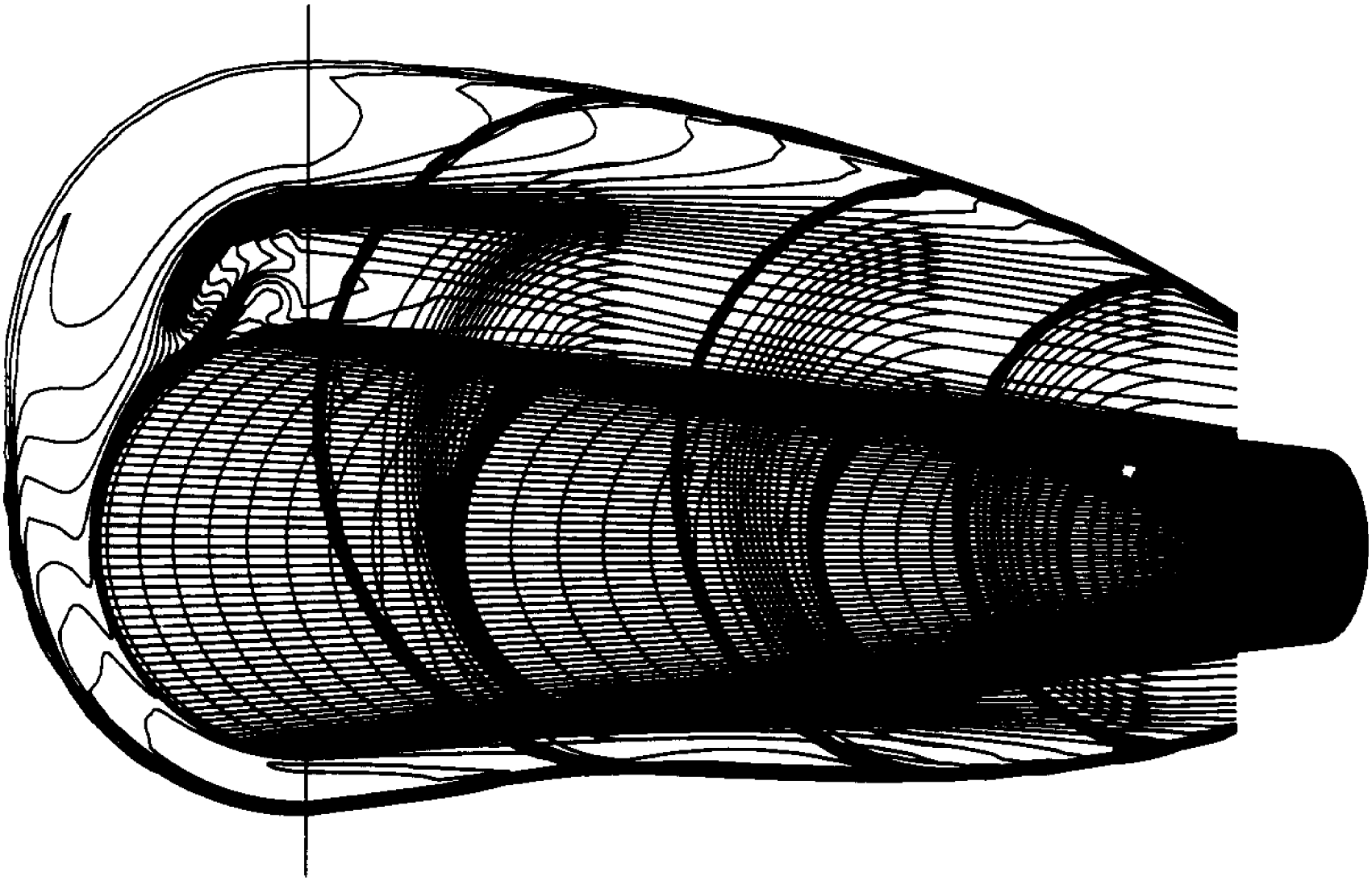


Figure 8.9: Mach number contours. (α= 5,° M_∞= 15, $Re_{\infty,L}$= 1.33x10^5, T_∞= 227K, 66x85 FIXED grid)

re-combine back to molecular oxygen, doing so more rapidly on the windward side of the cone than on the leeward side.

Figures 8.7 and 8.8 compare computed pressure contours for the solution-adapted and fixed grid results, respectively. Again, significant improvements in resolving the shock are noted for the solution-adapted case.

Figure 8.9 illustrate Mach number contours computed using a fixed grid containing 65 grid cells in the circumferential direction and 84 cells in the radial direction. Comparison with Fig. 8.3 shows that the solution-adapted case employing 65×64 grid cells provides a solution which is better resolved than the fixed grid employing 24% more grid points.

The solution-adapted case (66×65 grid) required 4500 seconds of CPU time on the Cray YMP/832 computer at Ames Research Center. The fixed grid solution of Fig. 8.9 (66×85 grid) required 6350 seconds of CPU time. The fixed grid case employing 66×65 grid points required 2800 seconds of CPU time. The differences in the CPU time of the solution-adapted case and the fixed grid case employing 66×65 grid points is partially attributed to an additional sub-iteration performed in the flow-chemistry coupling for the solution-adapted case. A total of five iterations of Eq. (8.2) were performed in the fixed grid cases while an additional iteration was required for the solution-adaptive case to bring the density residual down to a level equal to that of the fixed grid cases. Another reason for the reduction in computational speeds for the coarse grid solutions over the fine grid solution is vector length of 64 of the Cray YMP computer.

CHAPTER 9

Concluding Remarks

The work presented in the preceding chapters is now summarized. A brief overview of the development of the present solution-adaptive grid algorithm is first provided. Following the overview is a summary of the important contributions of the present work in the use of adaptive grids for improving numerical predictions of high-speed parabolic flows. Finally, suggestions for further improving the solution-adaptive grid procedures presented in this dissertation are provided.

9.1 Overview

A solution-adaptive grid algorithm has been developed and applied to a parabolized Navier-Stokes solver. The resulting solution-adaptive parabolized Navier-Stokes algorithm obtains numerical solutions for high-speed flow applications in the following manner. A preliminary grid plane is first constructed at a distance $\Delta\xi$ downstream of the current marching station. This is done by projecting the upstream grid plane at the current marching station downstream. Using the numerical integration techniques of Chapter 2, a preliminary solution is obtained on the preliminary grid plane. Based on this solution, gradients of selected flow-field variables are computed and used as an error measure or weighting function. Each coordinate line in each cross-flow direction is then adapted using the techniques of Chapters 3, 5 and 7. The grid adaptation step results in a re-distribution of grid points, clustering points in regions of high solution error such as boundary layers, shocks, expansion fans and shear layers. An improved solution for the flow-field is then obtained on this refined grid plane. Before continuing downstream to the next marching station, the grid-fitting algorithm is used to determine the farfield extent of solution variation. A new grid plane is then constructed with its outer boundary defined by the grid-fitting algorithm.

An adaptive grid algorithm was formulated in Chapter 3 which employs an equi-distribution concept for the reduction of overall solution error. Each computational coordinate line is modeled as a system of tensional springs with the spring constants set equal to a weighting function. To provide grid smoothness and enforce orthogonality, a torsion spring analogy is incorporated which relates neighboring grid lines and limits grid point movement. Weighting functions are chosen from a number of flow-field variable gradients and/or curvatures, thus allowing equal consideration to each during the adaptation process.

In Chapter 4, a number of two-dimensional flow-fields were investigated to demonstrate the improvements obtainable with this algorithm. The hypersonic flow over a leading-edge compression ramp was computed using both the solution-adaptive algorithm and using a fixed grid. The numerical resolution in the shock interaction region was shown to be significantly enhanced by the use of the solution-adaptive algorithm, and is even better than a fixed grid solution obtained with the use of twice as many grid points. Surface pressure and heat transfer predicted using the solution-adaptive algorithm also showed better agreement with experiments than predictions obtained using fixed grids.

The hypersonic flow through a cowl-type inlet channel was also investigated in Chapter 4. The flow-field structure predicted using the adaptive grid showed significant improvements in resolution over results using a fixed grid. Pressure predictions using the adapted grid also showed improved agreement with inviscid theory, further illustrating the benefits of the present algorithm.

The inviscid axi-symmetric flow over a cone-cylinder geometry was also computed. It was found that as many as three times the number of grid points were required when performing a fixed grid calculation to obtain the same level of agreement with experiment as the predictions obtained using the solution-adaptive algorithm. For the same level of accuracy, computational times were less when the solution-adaptive grid procedure was employed.

In Chapter 5, the grid adaptation methods presented in Chapter 3 were extended for application to three-dimensional flows. This extension involved the modification of the adaptation equations to include multiple torsional dependence to provide grid smoothness and orthogonality in relation to neighboring grid lines in both streamwise and cross-flow directions. Techniques were also presented in Chapter 5 to proportion the adapted cross-flow grid plane at the current marching station to the next marching station for use as a preliminary grid plane.

In Chapter 6 the hypersonic flow over a right-circular cone was numerically investigated at three different angles of freestream incidence. Due to the angle of incidence, a three-dimensional analysis was needed and grid adaptation was employed in both cross-flow directions. For each angle of incidence case, significant improvements in flow-field resolution were observed by the use of the adaptive grid. Pitot pressure profiles in the leeward symmetry plane predicted using the adaptive grid showed improved agreement with experimental results over the pressure predictions obtained using a fixed grid containing an equal number of points.

In Chapter 7 the solution-adaptive grid procedures were modified for application to more general three-dimensional configurations. These modifications included the formulation of a new technique to project the adapted mesh plane downstream to the next marching station in preparation for the next step. A three-dimensional grid-fitting algorithm was also developed which enabled the solution-adaptive algorithm to provide a smoother adapted grid. The torsional force term formulation of the system of grid adaptation equations was also modified in Chapter 7. This modification increased the usability of the algorithm by reducing the dependence of the user-selected adaptation parameters on the relative scales of the grid. This increased the range of usable parameters, thus reducing the number of trial solutions required for parameter adjustment.

The algorithm was then tested on the hypersonic flow over two lifting body configurations. The first was the Ames allbody vehicle, and flow-field results were computed at zero and 5° angle of incidence. Results computed using the solution-adaptive grid-fitting scheme were compared with those obtained using conventional fixed grids. Significant improvements were again observed in flow-field resolution as well as pitot pressure predictions. Results obtained using the grid-fitting scheme were also compared to results obtained without the use of grid-fitting. This comparison illustrated some of the problems encountered with

the adaptive grid algorithm when the freestream extent of the computational grid differs significantly in the circumferential direction.

The adaptive algorithm was also tested on the McDonnell Douglas generic option vehicle. The complexity of the surface grid of this geometry was the major reason for the reformulation of the downstream grid projection procedure of Chapter 7. A method of projecting the grid plane at the current marching station to the next marching station was required which could be applied to arbitrarily complex surface configurations. Flow-field results obtained using the adaptive algorithm illustrate the usefulness of the present grid adapting and generating techniques.

In Chapter 8, the solution-adaptive PNS solver was applied to the nonequilibrium flow of air over a blunt cone geometry. This application further illustrates the usability of the method for real flow situations. It was also demonstrated that significant computational savings can result with the use of the solution-adaptive grid algorithm.

9.2 Summary of Present Contributions

This section outlines a few of the important contributions of the present work to grid adaptation methods. As stated in Chapter 1, much work has been done in the past on adaptive grid methods. Davies and Venkatapathy (1989), Djomehri and Deiwert (1988), Gnoffo (1982) and Nakahashi and Deiwert (1984, 1985, 1986) have all made significant contributions to adaptation methods similar in their basic ideas to the present one, in that they employ a tension and torsion spring analogy to formulate the equations that yield the adapted grid point positions. The present work, however, makes a number of distinct and unique contributions which make the method different and better than other reported contributions. Some of the important contributions of the present work are given below.

Application to a parabolized Navier-Stokes solver for simultaneous solution and grid adaptation. The present solution-adaptive grid algorithm is embedded in a PNS solver. The resulting code obtains accurate numerical solutions to high-speed flow-fields in a single pass process. An improved solution is obtained on the refined grid immediately after grid adaptation is performed. In order for this concept to function properly, the grid must remain smooth at each step, maintaining a sufficient degree of alignment between the upstream and downstream sides of the computational slab.

Weighting functions selected from a user specified number of flow-field variables. The present technique of selecting weighting functions incorporates an algorithm whereby a number of flow-field variable gradients are first normalized. The largest normalized gradient is then chosen to represent the weighting function at each point along a coordinate line. Thus equal consideration is given to all variables when, for instance, pressure is changing rapidly in one region of the flow but may be unimportant in other regions, where other variables may be changing significantly. The present technique of selecting weighting functions considers all changes in all flow-field variables.

Grid-fitting procedure which reduces the number of grid points in the freestream and aligns the grid with flow-field structure. For complex three-dimensional problems it is often difficult to define a grid that is accurately aligned with the shock structure. If the grid is not aligned to some extent, grid adaptation using the present technique is very difficult. Grid lines tend to distort in the freestream as discussed in the results section of Chapter 7. The parabolized Navier-Stokes solver experiences difficulties when the shock speed relative to cross-flow grid lines is large. The grid-fitting algorithm eliminates these problems by accurately aligning the outer grid line with the outermost shock. As a result the shock remains stationary relative to the grid and, furthermore, a maximum

grid density is obtained in the region of interest. This feature also improves the performance of the adaptive grid algorithm, eliminating the severe grid distortions which appear without the use of the grid-fitting scheme.

Formulation of a single torsional force term in the adaptation equations instead of two terms as in previous formulations. Previous three-dimensional applications of the present adaptive grid technique [Nakahashi and Deiwert, (1986)], [Djomehri and Deiwert, (1988)] have incorporated the use of two torsional force terms in the grid adaptation equations. A user-specified parameter is required for each term. In Chapter 6 this technique was used without success to adapt the grid for the flow over cones at angles of incidence. It was difficult to find appropriate values for the λ torsion parameters which permitted an uninterrupted second solution step in the present solution-adaptive marching scheme. The torsional formulation presented in Chapter 5 was then devised in an attempt to provide a method which allowed a wider range of appropriate adaptation parameters. Chapter 5 presents a method whereby a single torsion parameter is specified and the torsional dependence is divided into two components by proportioning the relative magnitudes of the streamwise and cross-flow torsion terms (RHS Eq. 5.2) with the relative grid scales in the respective directions (See Eq. 5.3). Although this technique produces satisfactory results for relatively simple geometries such as cones, it was found to be inadequate for more complex situations such as those of Chapter 7. It was found that a suitable torsion relationship was not, in general, proportional to the relative grid scales in different directions.

In an attempt to further improve the relationship between neighboring grid lines in the adaptation equations, a different approach was devised in Chapter 7. This technique involves defining a single grid reference position, eliminating one of the torsional force terms in the adaptation equations altogether. This technique involves the re-inclusion of a second parameter (ϵ of Eq. 7.4) to proportion the

grid reference positions from the cross-flow and streamwise directions to define a single grid reference position. Thus, with this formulation, only one torsional force term appears in the adaptation equations (Eq. 7.2). This approach was found to be more robust in that a wider range of values for the torsional parameter, λ, exists which results in a smooth adapted grid for the second solution step of the solution-adaptive marching procedure.

9.3 Comments on Further Research

In the present work, no attempt was made to move the surface grid points. For all applications, the points at the surface remained fixed. One technique which could significantly enhance the effects of the present grid adaptation techniques is to provide a means for re-distributing the surface points. To devise an algorithm which would re-cluster points on the surface of an arbitrary geometry in a manner so as to preserve the cross-sectional shape is a formidable task. In the present work, when adapting in the circumferential direction, the first coordinate line of grid points on the surface was not adapted. Thus, when adapting the circumferential coordinate lines close to the surface, grid point movement was severely restricted due to the enforcement of orthogonality at the surface. As a result, reduction of overall solution error in the circumferential direction by the re-distribution of the points near the surface was compromised in order to enforce grid orthogonality in this region. By sliding the grid points along the surface cross-section, grid orthogonality can be maintained while simultaneously reducing solution error by the less restricted re-distribution of grid points.

Another means of further improvement and utility of the present methods is to devise a mechanism for introducing additional grid points in selected regions of the flow-field. This can be accomplished by flagging regions where the overall error has not been reduced to a prescribed level. Additional grid points can then be added to these regions for further solution enhancement.

Additional applications of interest include a three-dimensional internal flow whereby no-slip boundary conditions are enforced throughout. An application of this nature introduces the additional task of enforcing grid adaptation boundary criterion at more than one edge of the computational grid to ensure adequate maintenance of wall spacing. The implementation of this additional task is straight-forward, and is currently being investigated.

Bibliography

1. Anderson, D. A., Tannehill, J. C. and Pletcher, R. H. *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York, 1984.

2. Baldwin, B.S. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows." AIAA Paper 78-257, Jan. 1978.

3. Berger, M.J. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations." *AIAA Journal*, **23** (May 1985), pp. 561-586.

4. Bird, R.B., Stewart, W.E. and Lightfoot, E.N., *Transport Phenomena*, John Wiley & Sons, 1960.

5. Blottner, F.G., Johnson, M. and Ellis, M., "Chemically Reacting Viscous Flow Program for Multi-Component Gas Mixtures." Report No. SC-RR-70-754, Sandia Laboratories, Albuquerque, NM, Dec. 1971.

6. Buelow, P.E., Tannehill, J.C., Ievalts, J.O. and Lawrence, S.L., "A Three-Dimensional Upwind Parabolized Navier-Stokes Code for Chemically Reacting Flows." AIAA Paper 90-0394, 1990.

7. Brackbill, J.U., "Coordinate System Control: Adaptive Meshes." Numerical Grid Generation, Ed. J.F. Thompson, North-Holland, 1982.

8. Chakravarthy, S.R. and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws." AIAA Paper 85-0363, 1985.

9. Chakravarthy, S. R., Szema, K. Y. "Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets." *Journal of Aircraft*, **24** (February 1987), 73-83.

10. Davies, C.B. and Venkatapathy, E.W., "A Simplified Self-Adaptive Grid Method, SAGE." NASA TM-102198, October 1989.

11. Djomehri, D.J., and Deiwert, G.S., "Three-Dimensional Self-Adaptive Grid Method for Complex Flows." NASA TM-101027, November, 1988.

12. Dwyer, H.A., Kee, R.J. and Sanders. B.R., "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer." *AIAA Journal*, **18** (October 1980), pp. 1205-12.

13. Dwyer, H.A., "Grid Adaptation for Problems in Fluid Dynamics." *AIAA Journal* **22** (August 1984), pp. 955-60.

14. Edney, B., "Anomalous Heat Transfer and Pressure Distributions on Blunt Bodies at Hypersonic Speeds in the Presence of an Impinging Shock." The Aeronautical Research Institute of Sweden, Report 115, Stockholm, Feb. 1968.

15. Greenberg, J.B., "A New Self-Adaptive Grid Method." AIAA Paper 83-1934, 1983.

16. Gnoffo, P.A., "A Vectorized, Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Problems." AIAA Paper 82-1018, St. Louis, MO, June 1982.

17. Harten, A. "On a Class of High Resolution Total-Variation-Stable Finite-Difference Schemes." *SIAM Journal of Numerical Analysis*, **21** (1984), pp. 1-23.

18. Harvey, A.D., Acharya, S., Lawrence, S.L., and Cheung, S., "A Solution Adaptive Grid Procedure for High-Speed Parabolic Flow Solvers." *AIAA Journal*, **29** (August, 1991), pp. 1336-42.

19. Harvey, A.D., Acharya, S. and Lawrence, S.L., "A Solution-Adaptive Grid Procedure for the Three-Dimensional Parabolized Navier-Stokes Equations." AIAA Paper 91-0104, Reno, NV, January, 1991.

20. Hawken, D.F., "Review of Adaptive-Grid Techniques for Solution of Partial Differential Equations." University of Toronto, IAS, Review No. 46, December 1985.

21. Hoffmann, K. A. *Computational Fluid Dynamics for Engineers*, Engineering Education System, Austin TX, 1989.

22. Holden, M.S. and Moselle, J.R., "Theoretical and Experimental Studies of the Shock Wave-Boundary Layer Interaction on Compression Surfaces in Hypersonic Flow." CALSPAN Rept. AF-2410-A-1, Buffalo, NY, October 1969.

23. Kallinderis, I., "Adaptation Methods for Viscous Flows." Ph.D. Dissertation, Massachusetts Institute of Technology, 1989.

24. Lawrence, S.L., Tannehill, J.C. and Chaussee, D.S., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations." *AIAA Journal*, **22** (December 1984), pp. 1755-63.

25. Lawrence, S.L., "Application of an Upwind Algorithm to the Parabolized Navier-Stokes Equations." Ph.D. Dissertation, Iowa State University, 1987.

26. Lawrence, S.L., Tannehill, J.C. and Chaussee, D.S., "Application of an Upwind Algorithm to the Three-Dimensional Parabolized Navier-Stokes Equations." AIAA Paper 87-1112-CP, Honolulu, HA, June, 1987.

27. Lawrence, S.L., "Comparison of the UPS Code with Experimental Data for the McDonnell Douglas Generic Option Vehicle." Fifth National Aerospace Plane Technology Symposium, Paper No. 78, October 18-21, 1988.

28. Lockman, W.K., Lawrence, S.L. and Cleary, J.W., "Experimental and Computational Surface and Flow-Field Results for an All-Body Hypersonic Aircraft." AIAA Paper 90-3067, Portland, OR., August 1990. Also, to appear *Journal of Aircraft.*

29. Molvik, G.A. and Merkle, C.L., "A Set of Strongly-Coupled Upwind Algorithms for Computing Flows in Chemical Nonequilibrium." AIAA Paper 89-0199, Reno, NV, January 1989.

30. Moukalled, F.H., "Adaptive Grid Solution Procedure for Elliptic Flows." Ph.D. Dissertation, Louisiana State University, 1987.

31. Nakahashi, K. and Deiwert, G.S., "A Practical Adaptive-Grid Method for Complex Fluid-Flow Problems." NASA TM-85989, June 1984.

32. Nakahashi, K. and Deiwert, G.S., "A Self-Adaptive Grid Method with Application to Airfoil Flow." AIAA Paper 85-1525, Cincinnati, OH, July 1985.

33. Nakahashi, K. and Deiwert, G.S., "Three-Dimensional Adaptive Grid Method." *AIAA Journal*, **24** (June 1986), pp. 948-954.

34. Prabhu, D.K., Tannehill, J,C. and Marvin. J.G., "A New PNS Code for Chemical Nonequilibrium Flows." *AIAA Journal*, **26** (July 1988), pp. 808-815.

35. Rai, M.M. and Anderson, D.A., "Application of Adaptive Grids to Fluid-Flow Problems with Asymptotic Solutions." *AIAA Journal*, **20** (April 1982), pp. 496-502.

36. Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes." *Journal of Computational Physics*, **43** (1983), pp. 357-372.

37. Steger, J. L. and Warming, R. F. "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite-Difference Methods." *Journal of Computational Physics*, **40**, (April, 1981), pp. 263-293.

38. Thompson, J.F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations." *Applied Numerical Mathematics*, 1 (1985), pp. 3-27. North-Holland.

39. Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W., *Numerical Grid Generation, Foundations and Applications*, North-Holland, Amsterdam, 1985.

40. Tracy, R.R., "Hypersonic Flow over a Yawed Circular Cone." Memorandum No. 69, Graduate Aeronautical Laboratories, California Institute of Tech., Pasadena CA, 1963.

41. van Leer, B., Thomas, J.L., Roe, P.L. and Newsome, R.W. "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations." AIAA Paper 87-1104, June 1987.

42. Vigneron, Y.C., Rakich, J.C. and Tannehill, J.C., "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges." AIAA Paper 78-1137, Seattle, WA., July 1978.

43. Wilke, C.R., "A Viscosity Equation for Gas Mixtures." *J. Chem. Phys.*, 18 (April 1950) pp. 517-520.

Appendix A:

Eigenvalues and Eigenvectors of the D Matrix

The eigenvalues of

$$D = \frac{\partial \hat{\mathbf{P}}_i}{\partial \hat{\mathbf{E}}^*} = \left[\left(\frac{\kappa_x}{J}\right)\left(\frac{\partial \mathbf{E}_i}{\partial \hat{\mathbf{E}}^*}\right) + \left(\frac{\kappa_y}{J}\right)\left(\frac{\partial \mathbf{F}_i}{\partial \hat{\mathbf{E}}^*}\right) + \left(\frac{\kappa_z}{J}\right)\left(\frac{\partial \mathbf{G}_i}{\partial \hat{\mathbf{E}}^*}\right) \right]$$
$$= \left(\frac{\partial \hat{\mathbf{E}}^*}{\partial \mathbf{U}}\right)^{-1} \left[\left(\frac{\kappa_x}{J}\right)\left(\frac{\partial \mathbf{E}_i}{\partial \mathbf{U}}\right) + \left(\frac{\kappa_y}{J}\right)\left(\frac{\partial \mathbf{F}_i}{\partial \mathbf{U}}\right) + \left(\frac{\kappa_z}{J}\right)\left(\frac{\partial \mathbf{G}_i}{\partial \mathbf{U}}\right) \right]$$

are obtained by the solution to the problem

$$\det\left[\left(\frac{\kappa_x}{J}\right)\left(\frac{\partial \mathbf{E}_i}{\partial \mathbf{U}}\right) + \left(\frac{\kappa_y}{J}\right)\left(\frac{\partial \mathbf{F}_i}{\partial \mathbf{U}}\right) + \left(\frac{\kappa_z}{J}\right)\left(\frac{\partial \mathbf{G}_i}{\partial \mathbf{U}}\right) - \lambda A^* \right] = 0 \qquad (A.1)$$

where, as in Chapter 2, the generic inviscid flux, $\hat{\mathbf{P}}_i$, is used to denote $\hat{\mathbf{F}}_i$ when κ is replaced by η. Similarly, $\hat{\mathbf{G}}_i$ is obtained when κ is replaced by ζ. The solutions to Eq. $(A.1)$ for the eigenvalues and associated eigenvectors can be found in Lawrence, (1987).

Appendix B:

Derivation of Inviscid Flux Jacobians

The conserved variables can be expressed in vector form as

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{pmatrix} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{pmatrix} \tag{B.1}$$

The flux vectors $\mathbf{E}_i$, $\mathbf{F}_i$ and $\mathbf{G}_i$ defined by Eqs. (2.2) can be written in terms of the vector of conserved variables $U_1, U_2, \ldots, U_5$ and, after doing so in the following manner, are denoted as $\mathbf{E}_i(U_j)$, $\mathbf{F}_i(U_j)$ and $\mathbf{G}_i(U_j)$.

$$\mathbf{E}_i(U_j) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E_t + p)u \end{pmatrix} = \begin{pmatrix} U_2 \\ \frac{U_2^2}{U_1} + (\gamma - 1)[U_5 - \frac{1}{2}\frac{U_2^2 + U_3^2 + U_4^2}{U_1}] \\ \frac{U_2 U_3}{U_1} \\ \frac{U_2 U_4}{U_1} \\ \frac{U_2 U_5}{U_1}\gamma - \frac{1}{2}(\gamma - 1)\frac{U_2}{U_1^2}(U_2^2 + U_3^2 + U_4^2) \end{pmatrix}$$

$$\mathbf{F}_i(U_j) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E_t + p)v \end{pmatrix} = \begin{pmatrix} U_3 \\ \frac{U_3 U_2}{U_1} \\ \frac{U_3^2}{U_1} + (\gamma - 1)[U_5 - \frac{1}{2}\frac{U_2^2 + U_3^2 + U_4^2}{U_1}] \\ \frac{U_3 U_4}{U_1} \\ \frac{U_3 U_5}{U_1}\gamma - \frac{1}{2}(\gamma - 1)\frac{U_3}{U_1^2}(U_2^2 + U_3^2 + U_4^2) \end{pmatrix}$$

$$\mathbf{G}_i(U_j) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E_t + p)w \end{pmatrix} = \begin{pmatrix} U_4 \\ \frac{U_4 U_2}{U_1} \\ \frac{U_4 U_3}{U_1} \\ \frac{U_4^2}{U_1} + (\gamma - 1)[U_5 - \frac{1}{2}\frac{U_2^2 + U_3^2 + U_4^2}{U_1}] \\ \frac{U_4 U_5}{U_1}\gamma - \frac{1}{2}(\gamma - 1)\frac{U_4}{U_1^2}(U_2^2 + U_3^2 + U_4^2) \end{pmatrix}$$

The total inviscid flux in the κ-direction is

$$\begin{aligned} \hat{\mathbf{H}}_i(U_j) &= \frac{\kappa_x}{J}\mathbf{E}_i + \frac{\kappa_y}{J}\mathbf{F}_i + \frac{\kappa_z}{J}\mathbf{G}_i \\ &= \frac{\kappa_x}{J}\mathbf{E}_i(U_j) + \frac{\kappa_y}{J}\mathbf{F}_i(U_j) + \frac{\kappa_z}{J}\mathbf{G}_i(U_j) \end{aligned}$$

which is identical in form to Eqs. (2.5a). After substituting the above expressions for $\hat{\mathbf{E}}_i(U_j)$, $\hat{\mathbf{F}}_i(U_j)$ and $\hat{\mathbf{G}}_i(U_j)$ and adding, the following expression for $\hat{\mathbf{H}}_i$ is

obtained.

$$\hat{\mathbf{H}}_i = \begin{pmatrix} \frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4 \\ \frac{U_2}{U_1}[\frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4] + \frac{\kappa_x}{J}(\gamma-1)[U_5 - \omega\frac{1}{2}\frac{U_2^2+U_3^2+U_4^2}{U_1^2}] \\ \frac{U_3}{U_1}[\frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4] + \frac{\kappa_y}{J}(\gamma-1)[U_5 - \omega\frac{1}{2}\frac{U_2^2+U_3^2+U_4^2}{U_1^2}] \\ \frac{U_4}{U_1}[\frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4] + \frac{\kappa_z}{J}(\gamma-1)[U_5 - \omega\frac{1}{2}\frac{U_2^2+U_3^2+U_4^2}{U_1^2}] \\ [\frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4][\frac{U_5}{U_1}\gamma - \frac{1}{2}(\gamma-1)\frac{U_2^2+U_3^2+U_4^2}{U_1^2}] \end{pmatrix} \qquad (B.2)$$

To obtain the streamwise numerical flux $\hat{\mathbf{E}}^*$ from the generic numerical flux defined by Eq. $(B.2)$, κ is set equal to ξ and ω is defined by Eq. (2.8). To obtain the cross-flow numerical fluxes, κ is replaced by η when $\mathbf{H}_i$ is equal to $\hat{\mathbf{F}}_i$ and κ is replaced by ζ when $\hat{\mathbf{H}}_i$ is equal to $\hat{\mathbf{G}}_i$ and ω is set equal to unity. The notation $\mathbf{H}_i(\kappa)$ is merely a compact method of writing $\hat{\mathbf{E}}^*(\xi,\omega)$, $\hat{\mathbf{F}}_i(\eta)$ and $\hat{\mathbf{G}}_i(\zeta)$. Differentiating Eq. $(B.2)$ with respect to the dependent vector $\mathbf{U}$ the following generic expression is obtained for the Jacobian matrices

$$\frac{\partial\hat{\mathbf{H}}_i}{\partial U} = \begin{bmatrix} 0 & \frac{\kappa_x}{J} & \frac{\kappa_y}{J} & \frac{\kappa_z}{J} & 0 \\ \omega\frac{\kappa_x}{J}\psi_p - \frac{U_2}{U_1}\theta & \frac{\theta}{U_1} - \frac{\kappa_x}{J}\frac{U_2}{U_1}[\omega[(\gamma-1)-1] & \frac{\kappa_y}{J}\frac{U_2}{U_1} - \frac{\kappa_x}{J}\omega(\gamma-1)\frac{U_3}{U_1} & \frac{\kappa_z}{J}\frac{U_2}{U_1} - \frac{\kappa_x}{J}\omega(\gamma-1)\frac{U_4}{U_1} & \frac{\kappa_x}{J}\omega(\gamma-1) \\ \omega\frac{\kappa_y}{J}\psi_p - \frac{U_3}{U_1}\theta & \frac{\kappa_x}{J}\frac{U_3}{U_1} - \frac{\kappa_x}{J}\omega(\gamma-1)\frac{U_2}{U_1} & \frac{\theta}{U_1} - \frac{\kappa_y}{J}\frac{U_3}{U_1}[\omega(\gamma-1)-1] & \frac{\kappa_z}{J}\frac{U_3}{U_1} - \frac{\kappa_y}{J}\omega(\gamma-1)\frac{U_4}{U_1} & \frac{\kappa_y}{J}\omega(\gamma-1) \\ \omega\frac{\kappa_z}{J}\psi_p - \frac{U_4}{U_1}\theta & \frac{\kappa_x}{J}\frac{U_4}{U_1} - \frac{\kappa_x}{J}\omega(\gamma-1)\frac{U_2}{U_1} & \frac{\kappa_y}{J}\frac{U_4}{U_1} - \frac{\kappa_y}{J}\omega(\gamma-1)\frac{U_3}{U_1} & \frac{\theta}{U_1} - \frac{\kappa_z}{J}\frac{U_4}{U_1}[\omega(\gamma-1)-1] & \frac{\kappa_z}{J}\omega(\gamma-1) \\ \psi_p\theta - \frac{U_5}{U_1^2}\gamma\theta & \frac{\kappa_x}{J}h_t - \theta(\gamma-1)\frac{U_2}{U_1} & \frac{\kappa_y}{J}h_t - \theta(\gamma-1)\frac{U_3}{U_1} & \frac{\kappa_z}{J}h_t - \theta(\gamma-1)\frac{U_4}{U_1} & \gamma\frac{\theta}{U_1} \end{bmatrix} \qquad (B.3)$$

where

$$\theta = \frac{\kappa_x}{J}U_2 + \frac{\kappa_y}{J}U_3 + \frac{\kappa_z}{J}U_4$$

$$\psi_p = \frac{1}{2}(\gamma-1)\Big[\frac{U_2^2+U_3^2+U_4^2}{U_1^2}\Big]$$

$$h_t = \frac{U_5}{U_1}\gamma - \psi_p$$

and, as before, ω is defined by Eq. (2.8) when $\hat{\mathbf{H}}_i(\kappa)$ is equal to $\hat{\mathbf{E}}^*(\xi)$ (streamwise flux Jacobian) and ω is set equal to unity when $\hat{\mathbf{H}}_i(\kappa)$ is equal to $\hat{\mathbf{F}}_i(\eta)$ or $\hat{\mathbf{G}}_i(\zeta)$

(cross-flow inviscid Jacobians). The inviscid Jacobians of Eqs. (2.17) and (2.18) are then approximated as

$$\frac{\partial(\hat{\mathbf{F}}_i^1)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}}$$

$$\frac{\partial(\hat{\mathbf{F}}_i^1)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{F}}_i^1)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{F}}_i^1)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}}$$

and

$$\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}}$$

$$\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{G}}_i^1)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}} = \frac{1}{2}(I + \mathrm{sgn}D)\frac{\partial(\hat{\mathbf{H}}_i)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}}$$

Appendix C:

Viscous Flux Jacobians

The viscous flux vectors $\mathbf{F}_v$ and $\mathbf{G}_v$ of Eqs. (2.2) can be expressed as a single generic viscous flux vector as

$$\mathbf{H}_v = \frac{\mu J}{\mathrm{Re}_L} \begin{pmatrix} 0 \\ \cdots \\ a_1 u_\kappa + a_5 v_\kappa + a_7 w_\kappa \\ \cdots\cdots \\ a_5 u_\kappa + a_2 v_\kappa + a_6 w_\kappa \\ \cdots\cdots \\ a_7 u_\kappa + a_6 v_\kappa + a_3 w_\kappa \\ \cdots\cdots\cdots \\ \frac{1}{2}[a_1(u^2)_\kappa + a_2(v^2)_\kappa + a_3(w^2)_\kappa] \\ +a_5(uv)_\kappa + a_6(vw)_\kappa + a_7(uw)_\kappa + a_4 T_\kappa \end{pmatrix} \qquad (C.1)$$

where

$$\begin{aligned} a_0 &= \left(\frac{\kappa_x}{J}\right)^2 + \left(\frac{\kappa_y}{J}\right)^2 + \left(\frac{\kappa_z}{J}\right)^2 & a_4 &= \frac{1}{(\gamma-1)M_\infty^2 \mathrm{Pr}} a_0 \\ a_1 &= a_0 + \frac{1}{3}\left(\frac{\kappa_x}{J}\right)^2 & a_5 &= \frac{1}{3}\left(\frac{\kappa_x}{J}\right)\left(\frac{\kappa_y}{J}\right) \\ a_2 &= a_0 + \frac{1}{3}\left(\frac{\kappa_y}{J}\right)^2 & a_6 &= \frac{1}{3}\left(\frac{\kappa_y}{J}\right)\left(\frac{\kappa_z}{J}\right) \\ a_3 &= a_0 + \frac{1}{3}\left(\frac{\kappa_z}{J}\right)^2 & a_7 &= \frac{1}{3}\left(\frac{\kappa_x}{J}\right)\left(\frac{\kappa_z}{J}\right) \end{aligned} \qquad (C.2)$$

The viscous flux vector, $\mathbf{F}_v = \mathbf{H}_v$ when κ of Eqs. $(C.1)$ and $(C.2)$ is replaced by η and, similarly, $\mathbf{G}_v = \mathbf{H}_v$ when κ is replaced by ζ.

The viscous flux Jacobians are then obtained by differentiating Eq. $(C.1)$ with respect to, $\mathbf{U}$, the vector of conserved variables of Eq. $(B.1)$. The resulting generic Jacobian can be written as

$$\frac{\partial \hat{\mathbf{H}}_v}{\partial U} = \frac{\mu J}{\mathrm{Re}_L} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ -a_1\left(\frac{u}{\rho}\right) - a_5\left(\frac{v}{\rho}\right) - a_7\left(\frac{w}{\rho}\right) & a_1\left(\frac{1}{\rho}\right) & a_5\left(\frac{1}{\rho}\right) & a_7\left(\frac{1}{\rho}\right) & 0 \\ \cdots\cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ -a_5\left(\frac{u}{\rho}\right) - a_2\left(\frac{v}{\rho}\right) - a_6\left(\frac{w}{\rho}\right) & a_5\left(\frac{1}{\rho}\right) & a_2\left(\frac{1}{\rho}\right) & a_6\left(\frac{1}{\rho}\right) & 0 \\ \cdots\cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ -a_7\left(\frac{u}{\rho}\right) - a_6\left(\frac{v}{\rho}\right) - a_3\left(\frac{w}{\rho}\right) & a_7\left(\frac{1}{\rho}\right) & a_6\left(\frac{1}{\rho}\right) & a_3\left(\frac{1}{\rho}\right) & 0 \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix}$$

with

$$b_{51} = -a_1\left(\frac{u^2}{\rho}\right) - a_2\left(\frac{v^2}{\rho}\right) - a_3\left(\frac{w^2}{\rho}\right) - a_5\left(\frac{2uv}{\rho}\right) - a_6\left(\frac{2vw}{\rho}\right) - a_7\left(\frac{2uw}{\rho}\right) + M_\infty^2 a_4\left(\frac{\gamma\psi_p^2 - c^2}{\rho}\right)$$

$$b_{52} = -a_1\left(\frac{u}{\rho}\right) - a_5\left(\frac{v}{\rho}\right) - a_7\left(\frac{w}{\rho}\right) - \frac{\gamma}{\mathrm{Pr}}a_0\left(\frac{u}{\rho}\right)$$

$$b_{53} = -a_5\left(\frac{u}{\rho}\right) - a_2\left(\frac{v}{\rho}\right) - a_6\left(\frac{w}{\rho}\right) - \frac{\gamma}{\mathrm{Pr}}a_0\left(\frac{v}{\rho}\right)$$

$$b_{54} = -a_7\left(\frac{u}{\rho}\right) - a_6\left(\frac{v}{\rho}\right) - a_3\left(\frac{w}{\rho}\right) - \frac{\gamma}{\mathrm{Pr}}a_0\left(\frac{w}{\rho}\right)$$

$$b_{55} = \frac{\gamma}{\mathrm{Pr}}a_0\left(\frac{1}{\rho}\right)$$

The viscous Jacobian terms of the implicit algorithm are then given as

$$\frac{\partial(\hat{\mathbf{F}}_v)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k+1,l}} \qquad \frac{\partial(\hat{\mathbf{G}}_v)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l+1}}$$

$$\frac{\partial(\hat{\mathbf{F}}_v)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k+\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} \qquad \frac{\partial(\hat{\mathbf{G}}_v)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k,l+\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{F}}_v)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k,l}} \qquad \frac{\partial(\hat{\mathbf{G}}_v)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l}}$$

$$\frac{\partial(\hat{\mathbf{F}}_v)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k-\frac{1}{2},l}}{\partial \mathbf{U}_{k-1,l}} \qquad \frac{\partial(\hat{\mathbf{G}}_v)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}} = \frac{\partial(\hat{\mathbf{H}}_v)_{k,l-\frac{1}{2}}}{\partial \mathbf{U}_{k,l-1}}$$

Appendix D:

Calculation of Weighting Function Exponent

The calculation of B is found by a Newton-Raphson iterative procedure. The iterative procedure is performed until a value for B is found that produces a computed minimum Δs_i equal to the requested Δs_{MIN} The initial value of B is assumed to be equal to the computed value of B at the last line (or 1.0 for the first line). The value of B at the (n+1)th iteration is computed from $B^{(n+1)} = B^{(n)} + \Delta B^{(n)}$. If $|\Delta s_{MIN} - \min\Delta s_i{}^{(n)}|$ is small, an acceptable value of B has been found, and the iteration is complete. $\Delta B^{(n)}$ can be found from the basic definition of a derivative

$$\frac{\partial}{\partial B}\min(\Delta s_i) = \lim_{\delta B \to 0} \frac{(\Delta s_{MIN} - \min\Delta s_i{}^{(n)})}{\Delta B^{(n)}} \qquad (D.1)$$

As noted previously, Δs_i is a minimum when $w_i = 1 + A$, and by substituting this into Eq. (3.3) and differentiating, we obtain

$$\frac{\partial}{\partial B}\min(\Delta s_i) = \frac{s_{max}}{1 + A}\frac{\partial}{\partial B}\left(\frac{1}{\psi}\right) \qquad (D.2)$$

where

$$\psi = \sum_{l=1}^{n} \frac{1}{w_l}$$

and

$$\frac{\partial}{\partial B}\left(\frac{1}{\psi}\right) = -\frac{1}{\psi^2}\frac{\partial \psi}{\partial B}$$

Carrying out the differentiation of ψ *w.r.t.* B

$$\begin{aligned}
\frac{\partial \psi}{\partial B} &= \frac{\partial}{\partial B}\left(\sum_{l=1}^{n} \frac{1}{w_l}\right) \\
&= \sum_{l=1}^{n} \frac{\partial}{\partial B}\left(\frac{1}{w_l}\right) \\
&= \sum_{l=1}^{n} \frac{\partial}{\partial B}\left(\frac{1}{1 + A\bar{f}_l^B}\right) \\
&= -A\sum_{l=1}^{n} \frac{\bar{f}_l^B \log \bar{f}_l}{w_l^2}
\end{aligned}$$

Equation ($D.2$) can now be solved, and after substituting

$$\sum_{l=1}^{n}\left(\frac{1}{w_l}\right) = \frac{s_{max}}{\min(\Delta s_i)(1+A)}$$

we obtain

$$\frac{\partial}{\partial B}\min(\Delta s_i) = \frac{A(1+A)}{s_{max}}[\min(\Delta s_i)]^2 \sum_{l=1}^{n} \frac{\bar{f}_l^B log \bar{f}_l}{w_l^2} \qquad (D.3)$$

Equation ($D.3$) can then be used to obtain ΔB from Eq. ($D.1$).

Appendix E:

Letter of Permission

AIAA
American Institute of Aeronautics and Astronautics

September 11, 1991

Albert D. Harvey III
P.O. Box 25064
Baton Rouge, LA 70893

Dear Mr. Harvey,

AIAA grants permission for you to reprint material from 3 papers, co-authored by you, in your Ph.D. dissertation. Please credit AIAA with either "Copyright AIAA; reprinted with permission," at each figure or in one acknowledgement section.

The three papers are AIAA 91-1567, AIAA 91-0104, and AIAA 91-3237.

We would also appreciate receiving a copy of your dissertation to add to the collection of the AIAA Library and to announce in International Aerospace Abstracts.

In addition, please address all correspondence to our Copyright Permissions Department at the address below.

Sincerely,

Barbara Lawrence
Division Director,
Technical Information

TECHNICAL INFORMATION SERVICE ■ 555 West 57th Street ■ New York, NY 10019 ■ Phone: (212) 247-6500 ■ FAX: (212) 582-4861

Vita

Albert Destrehan Harvey III was born on September 20, 1959 in Houston Texas. After completing his high school education in 1977, he became employed in the offshore oil industry in the Gulf of Mexico. In September 1984 he received an Associate of Science degree in Petroleum Technology from Nicholls State University in Thibodaux, Louisiana. In December 1987 he received the Bachelor of Science in Mechanical Engineering from Louisiana State University. In January 1988, he began his graduate studies under the direction of Dr. Sumanta Acharya. During his entire Ph.D. program he was supported by a grant from the Applied Computational Fluids Branch at NASA Ames Research Center in Moffett Field California. He spent the year of August 1989 to August 1990 working at Ames Research Center under the technical direction of Dr. Scott Lawrence.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Albert Destrehan Harvey III

Major Field: Mechanical Engineering

Title of Dissertation: Solution-Adaptive Grid Procedures for High-Speed Parabolic Flows

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

November 8, 1991